EXHIBIT 5

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of:	Chu et al.
U.S. Patent No.:	6,683,858
Issue Date:	Jan. 27, 2004
Serial No.:	09/604,961
Filing Date:	Jun. 28, 2000
Title:	HYBRID SERVER ARCHITECTURE FOR MIXING AND NON-MIXING CLIENT CONFERENCING

Mail Stop Ex Parte Reexam
Central Reexamination Unit
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REQUEST FOR *EX PARTE* REEXAMINATION OF U.S. PATENT NO. 6,683,858 UNDER 35 U.S.C. § 302 AND 37 C.F.R. § 1.510

TABLE OF CONTENTS

155853739.8
155853739.12

155853739.8
155853739.12

TABLE OF AUTHORITIES

APPENDICES

APPENDIX	DESCRIPTION
A	U.S. Patent No. 6,683,858 (the "'858 patent")
B	'858 Patent File History
C	Declaration of James Bress
D	Cisco Claim Construction Brief
E	Microsoft Dictionary 4th
F	Paltalk Claim Construction Brief
G	U.S. Patent No. 7,079,495 ("Pearce")
H	U.S. Patent No. 6,418,125 ("Oran")
I	"Proposal of a Method of for Voice Stream Multiplexing for IP Telephony Systems" to Hoshi, et al. ("Hoshi")
J	"An RTP Payload Format for User Multiplexing", IETF Internet Draft, to Rosenberg, et al. ("Rosenberg")
K	U.S. Patent No. 6,327,276 ("Robert")
L	U.S. Patent No. 6,584,093 ("Salama")
M	U.S. Patent No. 6,141,597 ("Botzko")
N	U.S. Patent No. 6,006,253 ("Kumar")
O	U.S. Patent No. 6,697,476 ("O'Malley")
P	ITU-T Recommendation H.323 (11/96)
Q	Patent Owner Infringement Contentions
R	IEEE Conference Proceedings for the 1999 Internet Workshop cover
S	IEEE Conference Proceedings for the 1999 Internet Workshop LOC MARC record
T	U.S. Patent No. 7,158,491
U	U.S. Patent No. 8,189,592
V	U.S. Patent No. 6,993,021
W	U.S. Patent No. 6,704,311

X	U.S. Patent No. 6,542,504
Y	U.S. Patent No. 6,804,237
Z	ITU-T Recommendation H.323 (09/99)
AA	IETF RFC 2462
BB	Internet Archive Web Page for www.ietf.org/ids.by.wg/avt.html
CC	Internet Archive Web Page for http://www.ietf.org/internet-drafts/draft-ietf-avt-aggregation-00.txt

I. INTRODUCTION

Cisco Systems, Inc. ("Cisco" or "Requestor") requests reexamination under 35 U.S.C. § 302 and 37 C.F.R. § 1.510 of claims 1–10 of U.S. Patent No. 6,683,858 (the "'858 patent") (Appendix A) assigned to Paltalk Holdings, Inc. ("Paltalk" or "Patent Owner").

This request presents substantial new questions of patentability as to each of the challenged claims. The challenged claims combine nothing more than three well-known elements of audio conferencing in circuit-switched and packet-switched networks: sending multiplexed packets to IP telephony devices that have the capability to mix audio, mixing audio in a centralized server for telephony devices that do not have the capability to mix audio (e.g., conventional PSTN phones), and using an active speaker list during an audio conference. The claimed act of determining the mixing capability of a client device is not only a blatantly obvious implementation detail but was disclosed in the art prior to the '858 patent, including the H.323 standard.

Because this request presents new prior art references that have not previously been substantively considered in relation to the features of the claims, *ex parte* reexamination should be granted and each of the challenged claims canceled as unpatentable.

II. THE ’858 PATENT

A. Overview of the ’858 Patent

The ’858 patent “is directed to a hybrid server architecture … whereby mixing (e.g., PC-based clients) and non-mixing (e.g., phone) clients can simultaneously participate in a single audio conference application.” (Appx. A (’858 patent), 2:18–22.) As illustrated in Figure 1 depicted below, the system 100 “includes a plurality of PC-based clients 102 (shown as clients 102a–102n) which connect to a wide area network (e.g., the public Internet) 104.” (Appx. A (’858 patent), 3:60–63.) The system 100 “also includes a plurality of telephone-based clients 108 (shown as clients 108a–108n) which connect to the PSTN 110 (i.e., the circuit-switched network).” The PSTN “is connected to the service provider’s facilities (i.e., server 116) through a gateway 112 and the switch 114.” (Appx. A (’858 patent), 3:66–4:4.) The service provider’s server or multipoint control unit (MCU) 116 is connected to switch 114, which enables the MCU 116 “to receive audio packets from both PC-based clients 102 using, for example, the SIP protocol, as well as receive H.323 protocol packets from the telephone-based clients 108 who connect via gateway 112.” (Appx. A (’858 patent), 4:5–11.)

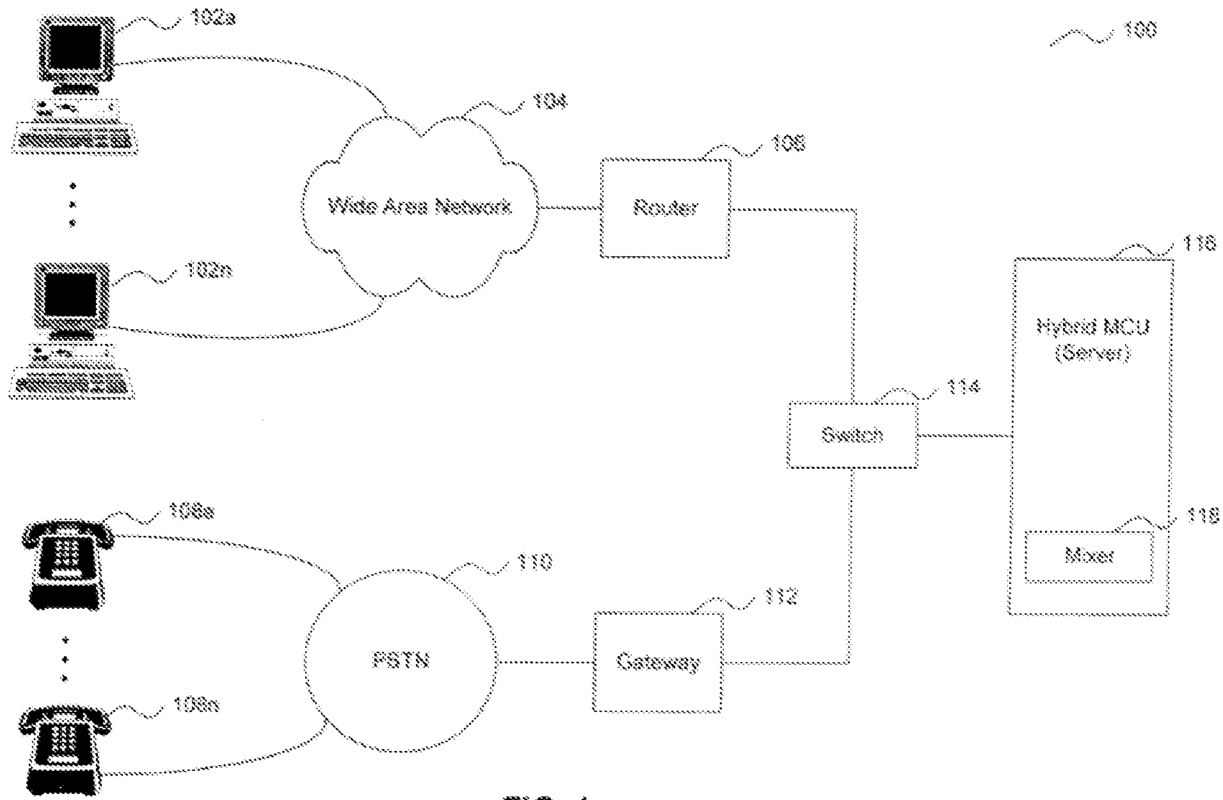

102a
100
104
108
Wide Area Network
Router
116
102n
Hybrid MCU
(Server)
114
Switch
106a
118
Mixer
110
112
PSTN
Gateway
106n

FIG. 1

in mixer 118 "forwards the packets created by mix/mux 208 to clients 102 and 108." (Appx. A ('858 patent), 4:55–56.)

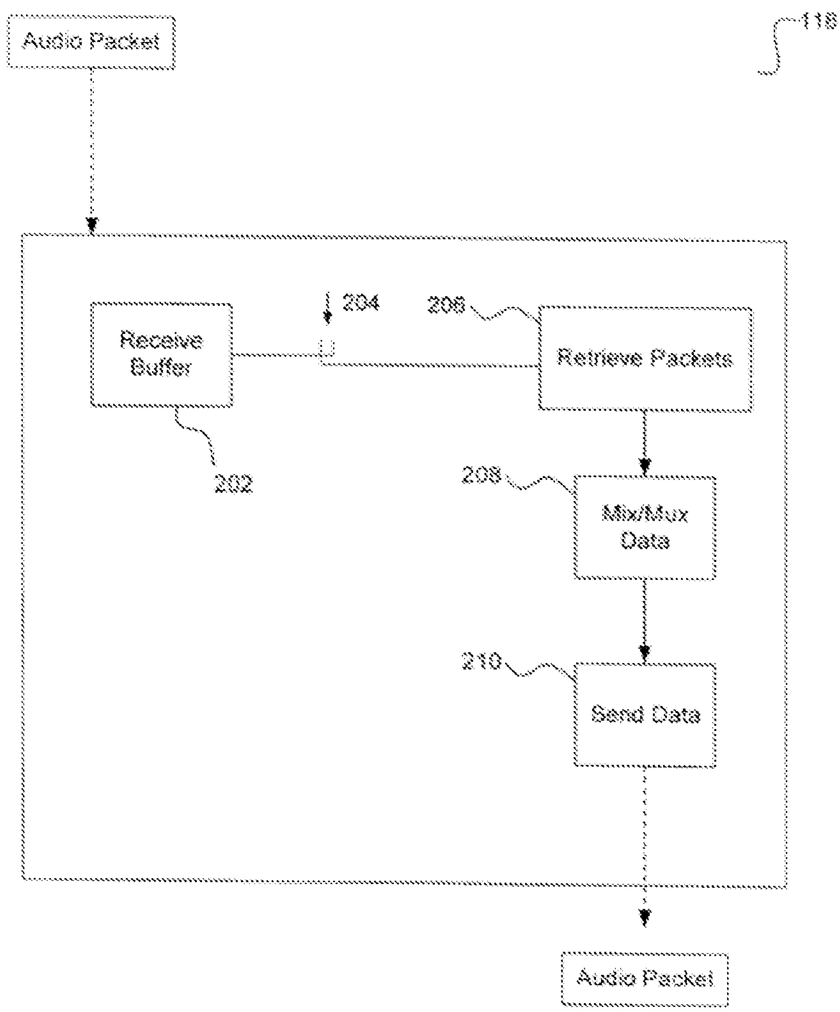


FIG. 2

The disclosed system performs the following general operational steps. First, upon detecting an event (e.g., "buffers 202 receive a pre-determined number of audio packets from each speaker"), the system "determines whether the active speaker list needs to be updated" and performs any necessary updates. (Appx. A

('858 patent), 4:32–5:23.) Second, the system determines "[w]hether a particular party is a mixing client (e.g., a PC-based client 102 using SIP) or not (e.g., a telephone client 108 using H.323)." (Appx. A ('858 patent), 5:24–35.) Third, the system "multiplexes (by employing the mix/mux 208) the audio stream data (stored on retriever 206) for all k active speakers" for mixing client. (Appx. A ('858 patent), 5:44–50.) For non-mixing clients, the active speaker audio data is decoded into raw uncompressed data, mixed, and encoded. (Appx. A ('858 patent), 5:53–63.) Finally, the system "either sends the multiplexed audio packet (created in step 314) to a mixing client or a mixed audio stream (created in step 320) to a non-mixing client." (Appx. A ('858 patent), 5:66–6:2.)

B. Prosecution History

U.S. Application No. 09/604,961—which issued as the '858 Patent—was filed on June 28, 2000. The Examiner issued a notice of allowance on September 18, 2003 without issuing a single office action. (*See* Appx. B ('858 patent file history), at PT0000083.) The prior art made of record—which the Examiner noted was not relied upon—consisted of two patents, including U.S. Patent No. 5,914,940 to Fukuoka, et al. and U.S. Patent No. 6,418,125 to Oran, which is addressed in detail below. (Appx. B ('858 patent file history), at PT0000084.) The Examiner did not identify or rely upon any other prior art references prior to issuing the notice of allowance. (Appx. B ('858 patent file history), at PT0000087.)

C. Level of Ordinary Skill in the Art

A person of ordinary skill in the art (a "POSITA") at the time of the patent would have had a bachelor's degree in electrical engineering, computer science, computer engineering, or another related field, and two to three years of experience working in the field of communication systems, hardware and software design, and network signaling services. (Appx. C (Bress Decl.), ¶22.)

D. Claim Construction

Terms in an *ex parte* reexamination are construed under the broadest reasonable interpretation standard ("BRI"). *In re Swanson*, 540 F.3d 1368, 1377–78 (Fed. Cir. 2008). In a co-pending district court litigation involving the '858 patent, the parties dispute the proper construction of the term "*multiplexed stream*" recited in independent claims 1 and 6 and dependent claims 2 and 7, "*PC-based equipment*" recited in dependent claims 4 and 9, and various means plus function limitations of claims 6–8 under the district court standard for claim construction. The disputed claim terms are presented below.

1. "*Multiplexed Stream*"[1] (claims 1, 2, 6, and 7)

While both parties agree that the plain and ordinary meaning applies for the term "*multiplexed stream*," the parties disagree on the plain and ordinary meaning.

[1] Claim language from challenged claims is indicated by italics herein.

The following table sets forth the positions of each party. (*See* Appx. D (Cisco Opening Claim Construction Brief), pp. 5–9.)

Patent Owner's Construction	Requestor's Construction
Plain and ordinary meaning	Plain and ordinary meaning, i.e., "a data structure containing a continuous sequence of interleaved packets of audio data from each client on the active speakers list"

Requester's construction reflects the plain and ordinary meaning of the term "*multiplexed stream*" and is therefore the proper interpretation under the broadest reasonable interpretation standard. A "*stream*" in data communications is commonly understood to be a continuous sequence of data. (*See* Appx. E (Microsoft Dictionary 4th), p. 424 (defining stream as "To transfer data continuously, beginning to end, in a steady flow"); Appx. D (Cisco Claim Construction Brief), p. 6.) In the context of the '858 patent specification and the '858 patent claims, multiplexing is when packets from two or more input sources are combined into a single output. (Appx. D (Cisco Claim Construction Brief), pp. 6–7.) Multiplexing creates a single output by interleaving packets from the input sources, one after the other. (Appx. D (Cisco Claim Construction Brief), pp. 6–7.) The claims impose such interleaving because they encompass the situation where only one packet is received from each client—meaning the multiplexing

necessarily interleaves the received packets from the clients. (Appx. D (Cisco Claim Construction Brief), pp. 6–7.) The result of multiplexing is therefore "*a sequence of interleaved packets of audio data from each client on the active speakers list*." Finally, because the claims recite that the "*multiplexed stream*" is sent to a "*first subset of the plurality of clients*," the "*multiplexed stream*" must be in a form capable of being transmitted—i.e., a data structure. (Appx. D (Cisco Claim Construction Brief), pp. 6–7.) This interpretation is consistent with the sole embodiment disclosed in the specification, which repeatedly refers to the output of the multiplexing step as a "multiplexed packet." (*See* Appx. A ('858 patent), 5:66–61 ("sends the multiplexed audio packet (created in step 314) to a mixing client"); 4:50–52 ("forms multiplexed audio packets"); 4:55–56 ("Mixer 118 also includes a packet sender 210 which forwards the packets created by mix/mux 208 to clients 102 and 108.").) A packet is therefore a data structure transmitted over a packet network. (Appx. D (Cisco Claim Construction Brief), pp. 6–7.)

Patent Owner has a broader view of the plain meaning of "*multiplexed stream*," unbounded by the specification or networked-based conferencing art. (*See* Appx. F (Paltalk Claim Construction Brief), pp. 3–10.) Under Patent Owner's interpretation, a "*multiplexed stream*" does not require a "data structure" or require the audio packets to be in a "continuous sequence." (*See* Appx. F (Paltalk Claim Construction Brief), pp. 7–8.) In fact, Patent Owner contends that the claimed

multiplexed stream can be "interrupted by other packets or even paused." (*See* Appx. F (Paltalk Claim Construction Brief), p. 8.) Patent Owner further disputes that claimed step of "multiplexing" requires input source data to be interleaved. (*See* Appx. F (Paltalk Claim Construction Brief), pp. 8–9.) Patent Owner even disputes the explicit language of independent claim 1, arguing that "*multiplexed stream*" does not require data from each client on the active speakers list. (*See* Appx. F (Paltalk Claim Construction Brief), p. 9.)

Although Patent Owner did not articulate the plain and ordinary meaning of the term, when its positions are considered as a whole, Patent Owner's plain and ordinary meaning appears to be simply "a flow of audio packets." This interpretation is simply not reasonable, even under a BRI standard.

2. PC-based equipment

Claims 4 and 9 each recite the term "*PC-based equipment*." The parties' positions in the co-pending district court litigation are set forth below.

Patent Owner's Construction	Requestor's Construction
"devices for personal computing"	Plain and ordinary meaning

(Appx. D (Cisco Claim Construction Brief), p. 9.)

3. Means plus function limitations

Claims 6–8 recite four limitations specified using "means plus function" language. The parties agreed to the construction for the "*means for storing*

information" and "*means for maintaining*" limitations of claim 6 as set forth in the following table:

Term	Parties' Agreed Construction
"means for storing information indicative of whether each of the plurality of clients has the capability to mix multiple audio streams"	**Function:** storing information indicative of whether each of the plurality of clients has the capability to mix multiple audio streams ------------------------------------ **Structure:** main memory 408 and processor 404 in Figure 4 as well as equivalents thereof
"means for maintaining a list of each of the plurality of clients that is an active speaker"	**Function:** maintaining a list of each of the plurality of clients that is an active speaker ------------------------------------ **Structure:** control logic, like that of control flow 300 in Figure 3, executed by the computer system 400 in Figure 4 as well as equivalents thereof

The parties' dispute regarding the "*means for removing*" recited in claims 7 and 8 centers on whether the structure for each is indefinite, as Requester contends

in the co-pending litigation. The following table sets forth the positions of each party:

Patent Owner's Construction	Requestor's Construction
Claim 7 **Function:** removing, before said packet sender sends said multiplexed stream to one of the plurality of clients which have the capability to mix multiple audio streams, from said multiplexed stream said packets of audio data received from said one of the plurality of clients, when said one of the plurality of clients is on said list of active speakers ------------------------------------ **Structure:** mixer 118 in Figures 1 and 2 as well as equivalents thereof	**Claim 7** **Function:** removing, before said packet sender sends said multiplexed stream to one of the plurality of clients which have the capability to mix multiple audio streams, from said multiplexed stream said packets of audio data received from said one of the plurality of clients, when said one of the plurality of clients is on said list of active speakers ------------------------------------ **Structure:** Indefinite

Patent Owner's Construction	Requestor's Construction
Claim 8	**Claim 8**
Function: removing, before said packet sender sends said combined packet to one of the plurality of clients which do not have the capability to mix multiple audio streams, from said combined packet said packets of audio data received from said one of the plurality of clients, when said one of the plurality of clients is on said list of active speakers ------------------------------------ **Structure:** mixer 118 in Figures 1 and 2 as well as equivalents thereof	**Function:** removing, before said packet sender sends said combined packet to one of the plurality of clients which do not have the capability to mix multiple audio streams, from said combined packet said packets of audio data received from said one of the plurality of clients, when said one of the plurality of clients is on said list of active speakers ------------------------------------ **Structure:** Indefinite

The language of claims 7 and 8 is clear and unambiguous. Packets of audio data received from one of the plurality of clients must be removed from either the "***multiplexed stream***" (Claim 7) or the "***combined packet***" (Claim 8) before either the multiplexed stream or combined packet is sent. That is, the multiplexed stream and combined packet are first created and then the "packets of audio data received from said one of the plurality of clients" are removed. The specification discloses no structure clearly linked to the removing function—regardless of whether it is removed from the multiplexed stream or the combined packet. (Appx. D (Cisco

Claim Construction Brief), p. 11.) The only description in the patent purporting to prevent a speaker from receiving their own audio does not include the audio packets from the active speaker client in the **creation** of either the multiplexed stream or combined packet. (*See* Appx. A ('858 patent), 5:48–52; 5:55–61.) That disclosure is different than a structure or algorithm that would remove the packets from a multiplexed stream or combined packet that were already created. (Appx. D (Cisco Claim Construction Brief), pp. 13–14.)

III. TECHNICAL REQUIREMENTS FOR *EX PARTE* REEXAMINATION UNDER 37 C.F.R. § 1.510

Requestor's *ex parte* reexamination request satisfies each requirement for *ex parte* reexamination of the '858 patent for the reasons set forth below.

A. 37 C.F.R. § 1.510(b)(1): Statement Pointing Out Each Substantially New Question of Patentability

A statement pointing out each substantial new question of patentability ("SNQ") based on the cited patents and printed publications in accordance with 37 C.F.R. § 1.510(b)(1) is provided below in §§IV and V.

B. 37 C.F.R. § 1.510(b)(2): Identification of Every Claim for which Reexamination is Requested

Reexamination is requested for claims 1–10 of the '858 patent in view of the prior art references discussed below. Specifically, Requestor asserts the following grounds of invalidity:

- Ground I: Claims 1 and 6 are obvious in view of Pearce in combination with Oran.
- Ground II: Claims 1, 5, 6, and 10 are obvious in view of the combination of Pearce and Oran in further combination with Hoshi.
- Ground III: Claims 1, 4–6, and 9–10 are obvious in view of the combination of Pearce and Oran in further combination with Rosenberg.
- Ground IV: Claims 2–3 and 7–8 are obvious in view of the combination of Pearce and Oran in further combination with Robert.
- Ground V: Claims 2–3 and 7–8 are obvious in view of the combination of Pearce, Oran, and Hoshi in further combination with Robert.
- Ground VI: Claims 2–3 and 7–8 are obvious in view of the combination of Pearce, Oran, and Rosenberg in further combination with Robert.
- Ground VII: Claims 4–5 and 9–10 are obvious in view of the combination of Pearce and Oran in further combination with Salama.
- Ground VIII: Claims 4–5 and 9–10 are obvious in view of the combination of Pearce, Oran, and Hoshi in further combination with Salama.
- Ground IX: Claims 4–5 and 9–10 are obvious in view of the combination of Pearce, Oran, and Rosenberg in further combination with Salama.
- Ground X: Claims 1–3, 5–8, and 10 are obvious in view of the combination of Butzko and Kumar.
- Ground XI: Claims 1–3, 5–8, and 10 are obvious in view of the combination of Butzko, Kumar, and Hoshi.
- Ground XII: Claims 1–10 are obvious in view of the combination of Butzko, Kumar, and Rosenberg.

C. 37 C.F.R. § 1.510(b)(2): Detailed Explanation of the Pertinency and Manner of Applying the Prior Art

A detailed explanation of the pertinency and manner of applying the prior art to every claim for which reexamination is requested in accordance with 37 C.F.R. § 1.510(b)(2) is provided below in §V.

D. 37 C.F.R. § 1.510(b)(3): Copy of Every Patent or Printed Publication Relied upon or Referred to

Requestor has provided a copy of every patent or printed publication relied upon or referred to in this *ex parte* reexamination request as Appendices A–CC. For convenience, the cited patents and printed publications are listed on Form PTO-1449, which is attached.

E. 37 C.F.R. § 1.510(b)(4): Copy of Entire Patent

Requestor has provided a copy of the entire ’858 patent as Appendix A in accordance with 37 C.F.R. § 1.510(b)(4).

F. 37 C.F.R. § 1.510(b)(5): Certification of Service

In accordance with 37 C.F.R. § l.510(b)(5), Requestor has provided below (*see* last page of this request) a certification that a copy of the request has been served in its entirety on the patent owner at the address as provided for in § l.33(c). The name and address of the party served has been indicated.

G. 37 C.F.R. § 1.510(b)(6): Certification of Third-Party Requestor

In accordance with 37 C.F.R. § l.510(b)(6), Requestor certifies that the statutory estoppel provisions of 35 U.S.C. § 315(e)(1) or 35 U.S.C. § 325(e)(1) do not prohibit Requestor from filing this *ex parte* reexamination request.

H. 37 C.F.R. § 1.510(a): Fee for Requesting Reexamination

The Patent and Trademark Office is authorized to charge Deposit Account No. 50-0665for the fee set forth in 37 C.F.R. § 1.510(a) for this Request and further authorizes payment for any additional fees to be charged to this Deposit Account.

I. Identification of Related Matters

Paltalk, the current Patent Owner, has asserted the ’858 patent in *Paltalk Holdings, Inc. v. Cisco Systems, Inc.*, 6:21-cv-00757 (W.D. Texas). A discussion of relevant claim construction disputes in this case is provided in Section §II.D.

IV. THE PRIOR ART RAISES SUBSTANTIAL NEW QUESTIONS OF PATENTABILITY

An overview of each of the prior art references that form the basis of the substantial new question of patentability are provided in the following sections:

Section V.A.1.a	U.S. Patent No. 7,079,495 to Pearce, et al. (“Pearce”)
Section V.A.1.b	U.S. Patent No. 6,418,125 to Oran (“Oran”)
Section V.B.1.a	“Proposal of a Method of for Voice Stream Multiplexing for IP Telephony Systems” to Hoshi, et al. (“Hoshi”)

155853739.8
155853739.12

Section V.B.1.b	"An RTP Payload Format for User Multiplexing", IETF Internet Draft, to Rosenberg, et al. ("Rosenberg")
Section V.C.1.a	U.S. Patent No. 6,327,276 to Robert, et al. ("Robert")
Section V.D.1.a	U.S. Patent No. 6,584,093 to Salama, et al. ("Salama")
Section V.E.1.a	U.S. Patent No. 6,141,597 to Botzko, et al. ("Butzko")
Section V.E.1.b	U.S. Patent No. 6,006,253 to Kumar, et al. ("Kumar")

Pearce, Hoshi, Rosenberg, Robert, Salama, Butzko, and Kumar were not before the Office during original prosecution of the '858 patent. As discussed in Section II.B, no Office Actions were issued during prosecution of the '858 patent. Therefore, although Oran is cited on the face of the patent, Oran was not applied in a rejection or discussed substantively on the record by the Examiner. Accordingly, none of the prior art grounds listed in Section III.B and discussed in Section V were previously considered by the Office during original prosecution.

Each of the combination listed in Section III.B presents a substantial new question of patentability of the challenged claims at least because (i) Pearce, Hoshi, Rosenberg, Robert, Salama, Butzko, and Kumar were not before the Office during original prosecution and (ii) these references and the proposed combinations have never been considered by the Office with respected the challenged claims in any proceeding. *See* M.P.E.P § 2242.

These new grounds disclose each and every limitation of challenged claims 1–10 as set forth in Section V. The new teachings set forth in the analysis of each

ground are such that a substantial likelihood exists that a reasonable examiner would have considered them important in determining whether or not the claims are patentable.

V. APPLICATION OF PRIOR ART TO CHALLENGED CLAIMS

The Request presents a detailed explanation of the pertinence and manner of applying the prior art references of Grounds I–XII to every claim for which reexamination is requested (claims 1–10) in this section. This analysis if further supported by Appendix C, a Declaration of James Bress, having over three decades of experience in the telecommunications industry. Each of these grounds presents new prior art combinations and arguments not previously considered by the Office.

A. Ground I: Claims 1 and 6 Are Obvious in View of Pearce in Combination with Oran

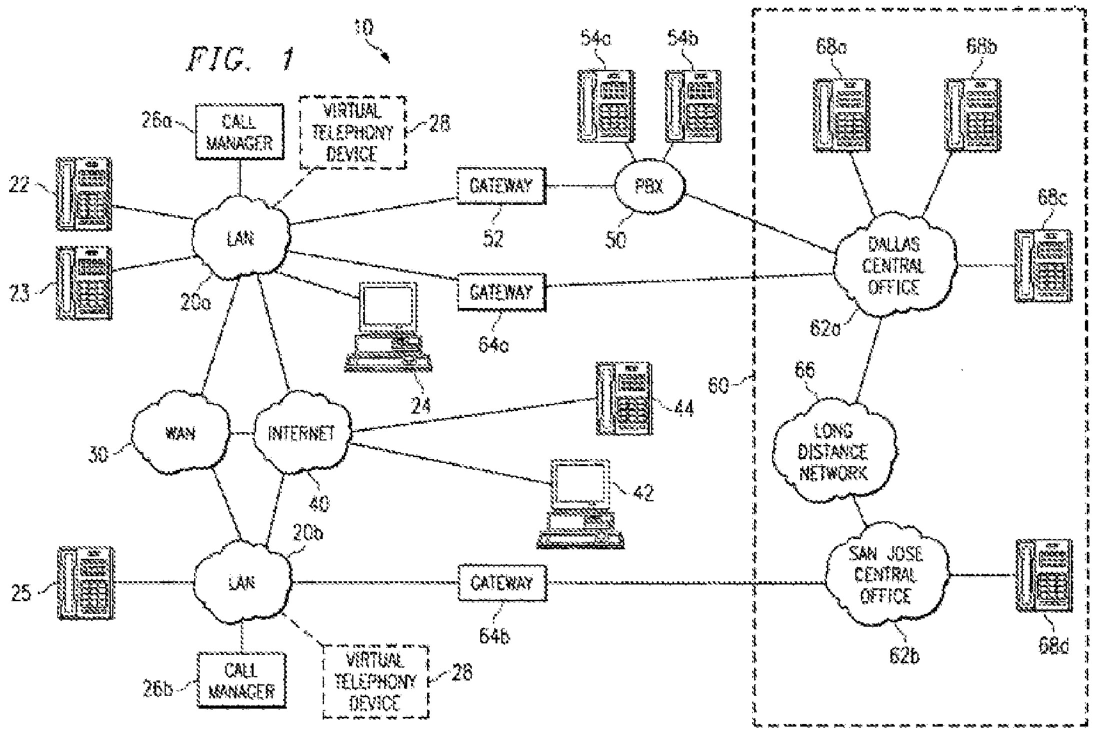
1. Overview of the Combination

a. Overview of Pearce

The application which issued as United States Patent No. 7,079,495 to Pearce, et al. ("Pearce"; Appx. G) was filed January 4, 2000 and issued July 18, 2006. Pearce is prior art under at least pre-AIA 35 U.S.C. § 102(e).

Pearce provides "a system and method that allow unicast telephony devices to effectively participate in a telecommunication session with multicast telephony devices." (Appx. G (Pearce), 2:11–14.) Figure 1 of Pearce, reproduced below with annotations, illustrates an exemplary communications network 10. As shown in

Figure 1, one or more IP telephony devices 22–24 are coupled to a LAN. The "IP telephony devices 22–24 have the capability of encapsulating a user's voice (or other media inputs) into IP packets so that the media can be transmitted over LAN 20 a, WAN 30 and/or Internet 40." (Appx. G (Pearce), 3:52–55.)



Pearce, Annotated Figure 1

One exemplary non-IP network "to which LANs 20 may be coupled is the Public Switched Telephone Network (PSTN) 60" including PSTN telephony devices 68. (Appx. G (Pearce), 3:32–34.) Calls between IP telephony devices (e.g., devices 22, 23) and non-IP telephony devices 68 "are made through a gateway" (e.g., gateway 64). (Appx. G (Pearce), 5:9–12.) The gateway 64 "converts analog

or digital circuit-switched data transmitted by PSTN 60 to packetized data transmitted by LAN 20, and vice-versa." (Appx. G (Pearce), 5:14–16.) For example, because "the digital format for voice transmissions over an IP network is often different than the format used on the digital trunks of PSTN 60, gateway 64 provides a conversion between these different digital formats, referred to as transcoding." (Appx. G (Pearce), 5:20–24.)

Pearce's network also includes a call manager 26a which "controls IP telephony devices 22-24." (Appx. G (Pearce), 4:7–8.) The call manager 26a "is an application that controls call processing, routing, telephone features and options (such as call hold, call transfer and caller ID), device configuration, and other telephony functions and parameters within communication network 10." (Appx. G (Pearce), 4:9–13.) Call manager 26a "can control all of the IP telephony devices on LAN 20a" and "it may also control IP telephony devices located across WAN 30," eliminating the need for a second call manager 26b. (Appx. G (Pearce), 4:13–18.)

"When a user wishes to place a call from one IP telephony device on LAN 20a to another IP telephony device on LAN 20a (an intra-LAN call), the originating telephony device transmits a signal to call manager 26a indicating the desired function and the telephony device to be called." (Appx. G (Pearce), 4:19–23.) The call manager 26a "checks on the availability of the target telephony device and, if available, sets up the call by instructing the originating telephony

device to establish a media stream with the target telephony device." (Appx. G (Pearce), 4:23–27.) After the call is initiated, a codec (coder/decoder) in the IP telephony device converts the voice "signals generated by the users of the telephony devices from analog voice signals into digital form." (Appx. G (Pearce), 4:46–49.) "The digitally encoded data is then encapsulated into IP packets so that it can be transmitted over LAN 20a." (Appx. G (Pearce), 4:53–55.)

For calls to a PSTN telephony device 68 from an IP telephony device 22 on LAN 20a, the voice signal "generated by the user of IP telephony device 22 is digitized and encapsulated" and "transmitted over LAN 20a to gateway 64." (Appx. G (Pearce), 5:34–37.) Gateway 64 "converts the data to the format (either digital or analog) used by the PSTN trunk to which the gateway is connected" and the voice signals are "sent to PSTN telephony device 68 over PSTN 60." (Appx. G (Pearce), 5:43–47.) "This process, and the reverse process, is continued between PSTN 60 and LAN 20a through gateway 64 until the call is complete." (Appx. G (Pearce), 5:47–49.)

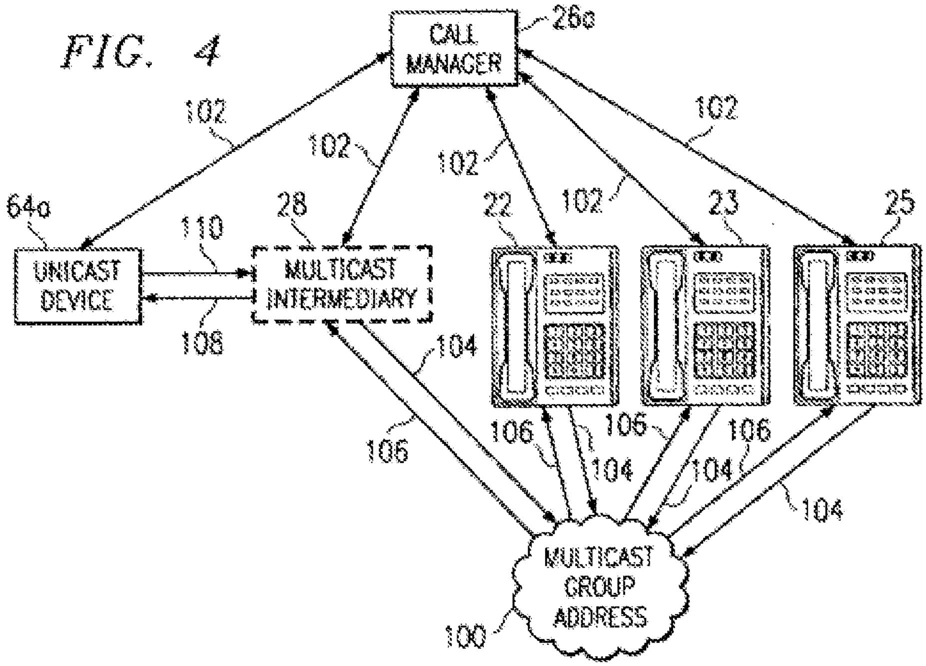
Pearce explains that "[t]here are three basic ways to transmit identical data to multiple receivers on a packet-based network: broadcast, unicast, and multicast." (Appx. G (Pearce), 9:5–7.) "A broadcast is a single data stream sent from a single device to every device on a network (or subnet)." (Appx. G (Pearce), 9:7–9.) In unicast, a source device "transmit[s] a unicast to each intended destination device."

155853739.8
155853739.12

(Appx. G (Pearce), 9:13–15.) "A multicast is a single data stream that is intended only for particular devices that have joined an appropriate 'multicast group.'" (Appx. G (Pearce), 9:23–25.) In multicast, "the source device generates a single data stream." (Appx. G (Pearce), 9:25–26.) To send IP multicast packets, the source "specifies a destination address that represents the multicast group" (e.g., a multicast group address). (Appx. G (Pearce), 9:39–43.) To receive multicast packets, an application on a device "requests membership in the multicast group." (Appx. G (Pearce), 9:43–45.)

Pearce describes a technique to permit a unicast telephony device to effectively participate in a multicast telecommunication session with one or more multicast devices. (*See, e.g.,* Appx. G (Pearce), 10:35–12:40.) Figure 4 of Pearce, reproduced below, illustrates a conference call (telecommunication session) between IP telephone devices 22, 23, 25 and unicast device 64a which acts as a gateway for PSTN telephony device 68a (not shown). (*See* Appx. G (Pearce), 10:43–61.)

FIG. 4



In this example, when gateway 64a sends a call initiation request to call manager 26a indicating that PSTN telephony device 68a requests a telecommunication session (e.g., a conference call) with telephony devices 22, 23, and 25, call manager 26a initiates the session between the devices that are available to participate. (Appx. G (Pearce), 10:55–67.) Based on "registration information previously sent by telephony devices 22, 23 and 25 to call manager 26a, call manager 26a determines that telephony devices 22, 23 and 25 support multicast communication" and "establishes a multicast group having a multicast group address 100." (Appx. G (Pearce), 11:1–6.) Call manager 26a further "instructs telephony devices 22, 23 and 25 to initiate outgoing media streaming 104 to multicast address 100, and to monitor multicast address 100 for incoming media streaming 106." (Appx. G (Pearce), 11:6–9.)

Call manager 26a further "determines from either registration information or the call initiation request that telephony device 64a is a unicast telephony device." (Appx. G (Pearce), 11:12–14.) A unicast telephony device is not "able to participate in a multicast telecommunication session because unicast telephony devices are not capable of monitoring a multicast group address to determine if any messages are being sent to the multicast group." (Appx. G (Pearce), 10:17–21.) A multicast intermediary can be "inserted into a telecommunication session on behalf of the unicast telephony device(s)" so that "the unicast telephony device(s) can effectively participate in a multicast telecommunication session." (Appx. G (Pearce), 10:31–34.) Accordingly, in the example of Figure 4, call manager 26a "generates a multicast intermediary 28" for unicast telephony device 64a. (Appx. G (Pearce), 11:15–16.) "As with telephony devices 22, 23 and 25, call manager 26a instructs multicast intermediary 28 to monitor multicast address 100 for incoming media streaming 106" and "signals telephony device 64a to initiate outgoing media streaming 110 to the logical port of multicast intermediary 28 that call manager 26a associated with multicast address 100." (Appx G (Pearce), 11:19–25.)

The multicast intermediary performs a number of functions in the multicast telecommunications session. First, "[w]hen multicast intermediary 28 receives incoming media streaming 110 from telephony device 64a, it notes the source address in the incoming packets, modifies the source address to the logical port of

155853739.8
155853739.12

multicast intermediary 28 that is associated with telephony device 64a, and forwards the packets as media streaming 104 to multicast address 100 using a communication module." (Appx. G (Pearce), 11:29–36.) Second, "multicast intermediary 28 forwards media streaming 108 that it receives from multicast address 100 to telephony device 64a as media streaming 108 using the communication module." (Appx. G (Pearce), 11:37–43.) Third, because "telephony device 64a is not directly capable of sending and receiving multicast messages," "multicast intermediary will typically **sort or mix media streaming 104 received from telephony devices**[2] 22, 23 and 25." (Appx. G (Pearce), 11:43–46.) Finally, the "[m]ulticast intermediary may also perform any other type of processing to convert media streaming 106 received from multicast group address 100 into a format appropriate for a unicast telephony device." (Appx. G (Pearce), 11:59–62.)

b. Overview of Oran

The application which issued as United States Patent No. 6,418,125 ("Oran"; Appx. H) was filed June 18, 1998 and issued July 9, 2002. Oran is prior art under at least pre-AIA 35 U.S.C. §102(e).

Oran's invention is directed "to managing audio packets in a packet-based audio system with multiple speakers and more particularly to adaptively selecting which audio packets to mix together as an audio signal depending on the active

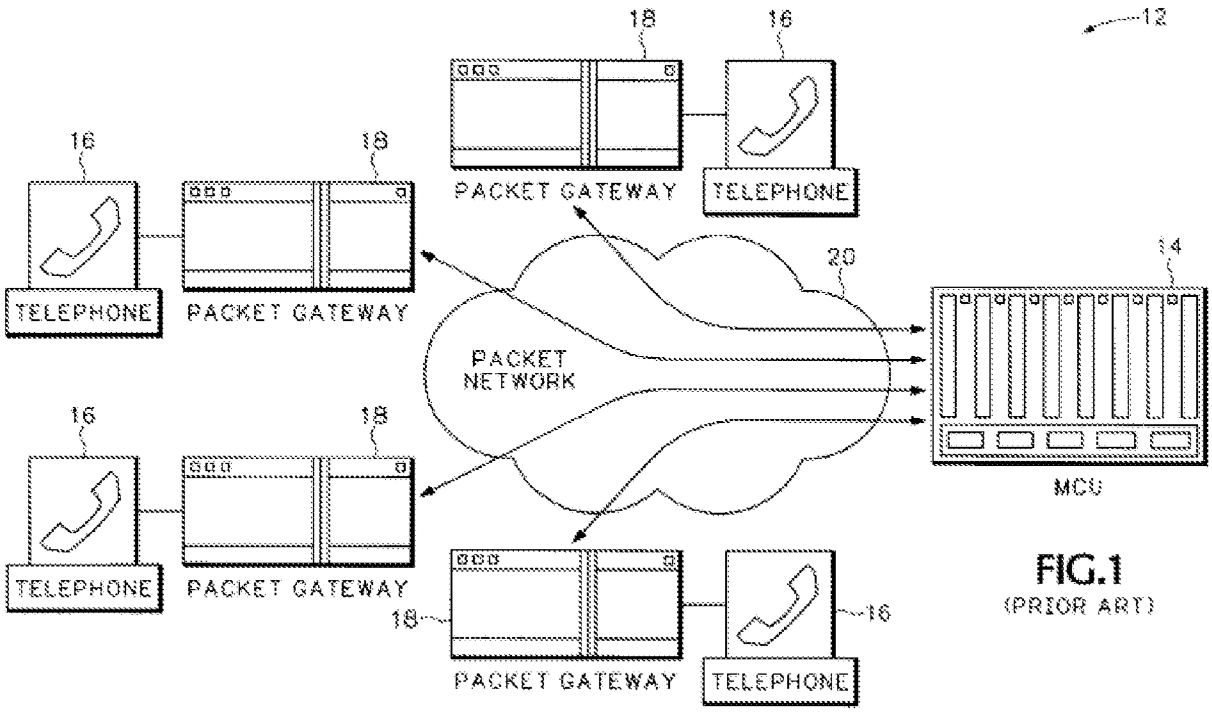
[2] Unless otherwise noted, all emphasis in bold and/or underlining added.

status of the multiple speakers." (Appx. H (Oran), 1:8–12.) Oran explains that in audio applications "there is often the need to talk with more than one speaker at a time." (Appx. H (Oran), 1:13–14.) Oran notes that in a pure "circuit-switched system, such as the Public Service Telephone Network (PSTN), these functions are typically handled either by an edge switch or a special purpose device called an 'audio bridge' or Multipoint Control Unit (MCU)." (Appx. H (Oran), 1:16–20.) According to Oran, in a packet audio system there are, "better solutions to the transport of audio, such as using multicast transmission." (Appx. H (Oran), 1:21–22.) However, these packet-based techniques "require the receivers to perform many of the processing functions of an MCU." (Appx. H (Oran), 1:23–24.)

(i) Prior-art Centralized Conference Server

Figure 1 of Oran (reproduced below) depicts a prior art method of managing audio for teleconferencing in a packet network. Oran's prior art system 12 includes conventional PSTN telephones 16 and associated gateways 18 that "perform the conversion between data packets containing audio data (audio packets) and audio signals passing to and from the telephones 16." (Appx. H (Oran), Figure 1, 2:54–56.) As shown in Figure 1, the centralized MCU 14 receives audio packets from the gateways over the packet network 20. A POSITA would understand that Figure 1 depicts the gateways "unicasting" their packets to the centralized MCU. (Appx. C (Bress Decl., ¶44.) After processing, MCU 14 "sends the resulting

mixing/selected data stream to packet gateways 18" for transmission to telephones 16. (Appx. H (Oran), 2:50–51.)



Oran, Figure 1

(ii) Active Speaker Management

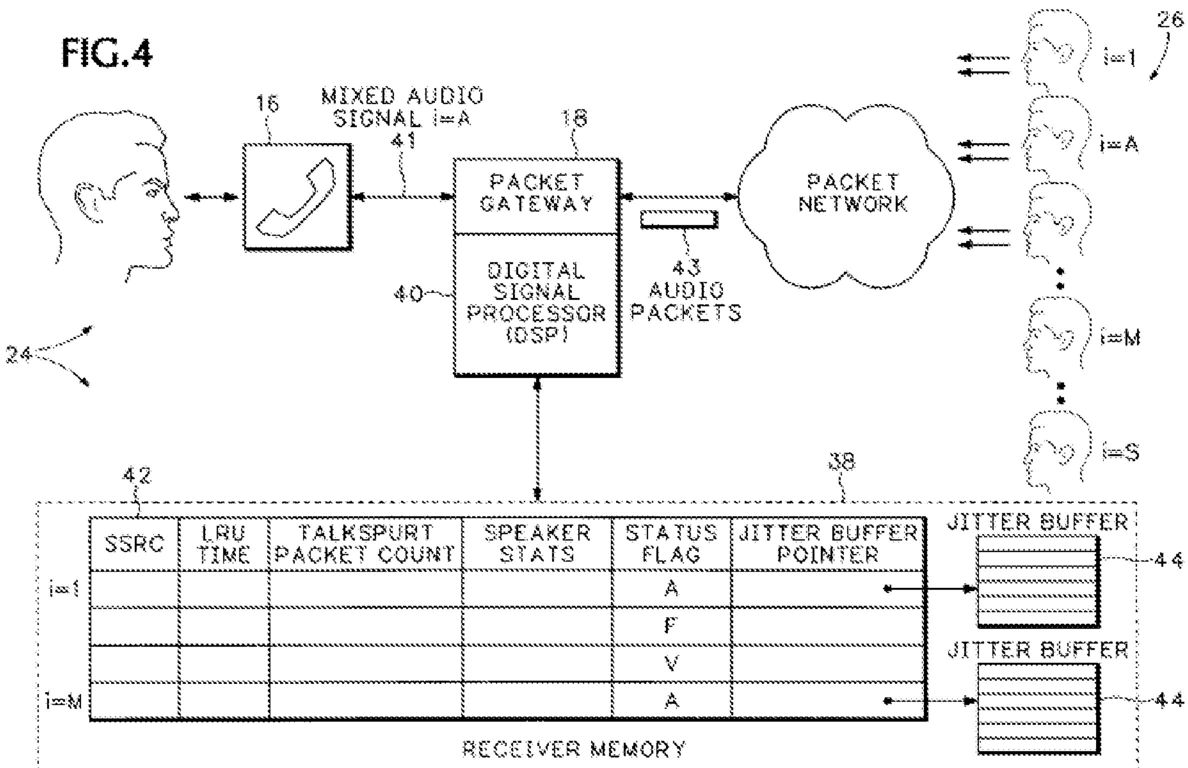
Oran describes "an adaptive speaker management scheme to intelligently select which speaker states and audio to retain and process." (Appx. H (Oran), 2:14–16.) In Oran's speaker management scheme, a receiver "receives audio packets and identifies speakers 26A-26D associated with the audio packets." (Appx. H (Oran), 3:40–42.) The receiver then "decides which speakers are currently 'active' and warrant keeping information about." (Appx. H (Oran), 3:43–44.) Oran's receiver "retains information on selected speakers that appear to be

actively participating in the telephone conference." (Appx. H (Oran), 3:58–60.) Of the speakers identified as "active," a receiver "independently decides which speakers to mix together to produce an output audio stream." (Appx. H (Oran), 3:61–63.)

In Oran, "[t]he speaker management scheme assumes reliable indication of which speakers 26A-26D (FIG. 2) are associated with which audio packet, independent of whether it arrives via unicast or multicast." (Appx. H (Oran), 4:14–17.) Oran explains that "for any given audio session, there is some number of potential speakers 26" which Oran calls "value S." (Appx. H (Oran), 4:31–32.) As illustrated in Figure 4 (reproduced below), a receiver has "enough memory 38 to keep state information in data array 42 for some subset of these speakers 26" which Oran calls "value M." (Appx. H (Oran), 4:32–35.) Each row in the "data array 42 represents a speaker entry containing the state information for one of the M speakers." (Appx. H (Oran), 5:9–10.) SSRC "is used to match speaker entries in the data array 42 with audio packets 43." (Appx. H (Oran), 5:11–12.) Speaker Stats "contain a variety of information about a speaker 26 used for sizing and operating a jitter buffer 44 when the speaker 26 is active." (Appx. H (Oran), 5:28–30.) The Status Flag of "V" "is used to identify either an inactive speaker or a speaker who is being ignored in favor of another speaker who is active" and a Status Flag of "F"

"identifies a speaker free to be used or reused by another speaker 26." (Appx. H (Oran), 5:40–46.)

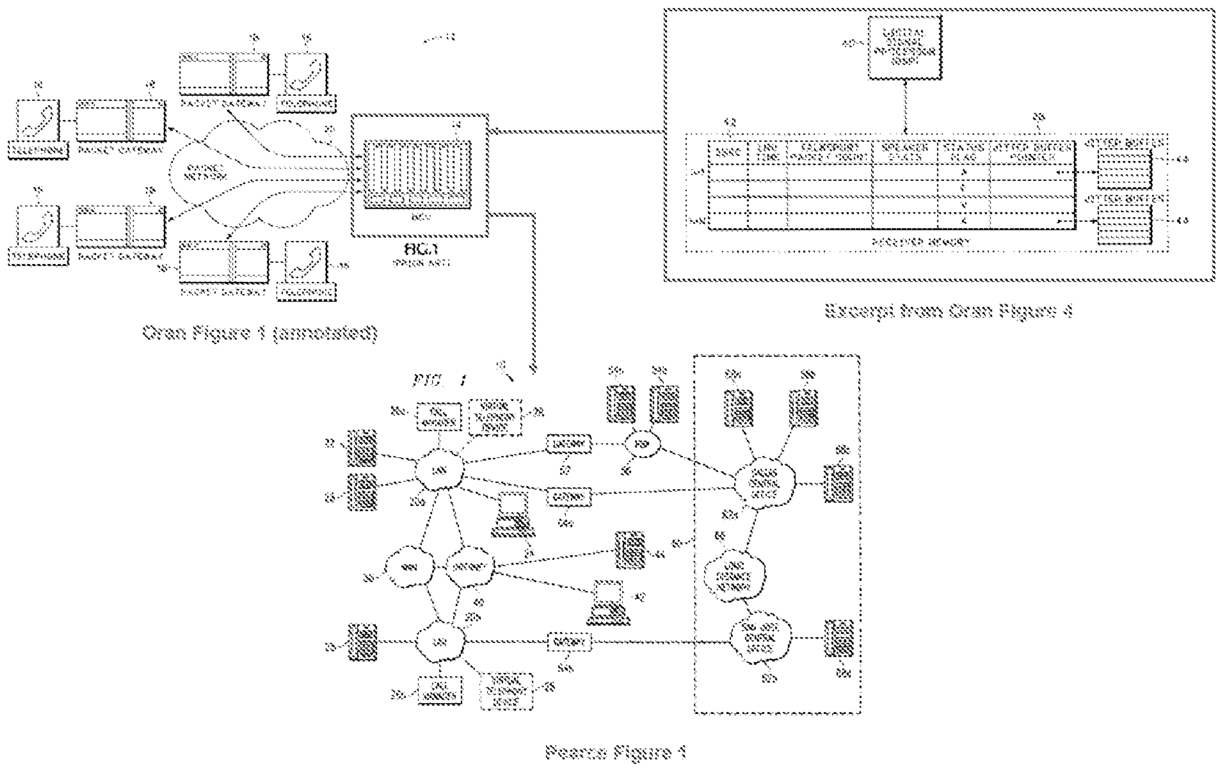
FIG.4



c. Motivation to Combine

A POSITA would have been motivated to combine the teachings of Pearce and Oran as illustrated in Bress Figure A below. First, a POSITA would have been motivated to utilize the Oran teachings of a prior-art centralized MCU with Pearce's multicasting system. In the combined system, Pearce's IP telephony devices and multicast intermediaries 28 (shown as a virtual telephony device in Pearce Figure 1) continue to act as multicast receivers. (Appx. C (Bress Decl.), ¶48.) These devices in the combination transmit their audio IP packets to the MCU

as unicast packets as taught by Oran's Figure 1 architecture. (Appx. C (Bress Decl.), ¶48.) The MCU, in turn, uses multicast to send audio packets for the telecommunication session. (Appx. C (Bress Decl.), ¶48.) As multicast recipients, the IP telephony devices and multicast intermediaries monitor the multicast address for packets. (Appx. C (Bress Decl.), ¶48.)



Bress Figure A

A POSITA would have been motivated to combine Oran's prior-art centralized MCU with Pearce's teachings of multicast transmission for conference calls for a number of reasons. (*See generally* (Appx. C (Bress Decl.), ¶¶ 50–53.) First, a POSITA would recognize that client-based conferencing systems have a drawback of requiring each client to have the requisite equipment and software to

receive and process multiple streams of audio packets. (Appx. C (Bress Decl.), ¶51, *citing*, e.g., Appx. K (Robert), 4:30–40.) A POSITA would have therefore been motivated to use a centralized conferencing platform to perform specific functions during a conference to permit clients with different capabilities to participate in a conference. (Appx. C (Bress Decl.), ¶52.) Second, Oran suggests the use of multicast transmission from a centralized MCU, noting that multicast packets can be used "for outbound transmission from the MCU." (Appx. H (Oran), 2:57–60.) Indeed, platforms using multicast for transmitting audio from a centralized platform to conference participants were well-known before the filing date of the '858 patent. (*See* Appx. M (Botzko), 6:4–20.)

Third, a POSITA would understand that a centralized server (e.g., a server hosting an MCU application) has far superior processing resources than individual telephony devices. (Appx. C (Bress Decl.), ¶52.) The increased processing capability for all participants in a conference permits the MCU to provide enhanced functionality within a conference call (e.g., active speaker management) which may not be possible or practical at every telephony device participating in a conference call. (Appx. C (Bress Decl.), ¶¶52, 61.) Additionally, a centralized MCU provides a consistent level of service to all participants regardless of their processing capabilities. (Appx. C (Bress Decl.), ¶¶52, 62.) IP telephony devices in a call may have limited processing capabilities whereas other IP telephony devices

may have faster processors. (Appx. C (Bress Decl.), ¶52.) Oran confirms that receivers may have "limited processing resources." (*See* Appx. H (Oran), 1:25–26.) Centralizing MCU functionality allows the conference call provider to provide the same level of service and consistent functionality to all conference call participants. (Appx. C (Bress Decl.), ¶¶52, 62.) For example, if functions such as active speaker identification are performed centrally rather than locally, each conference call participant will hear the same audio associated with the conference. (Appx. C (Bress Decl.), ¶62.) If active speaker identification is performed locally, the potential exists that different conference participants will hear different "versions" of the same call. (Appx. C (Bress Decl.), ¶62.)

A POSITA would be motivated to integrate Oran's active speaker management teachings into the centralized MCU, as illustrated in Bress Figure A (above). (*See generally* Appx. C (Bress Decl.), ¶¶ 58–62.) Oran describes a "prior art MCU-based system 12," wherein an "MCU 14 takes the responsibility of listening to all speakers that may talk via telephones 16 and performs all mixing and speaker selection." (Appx. H (Oran), 2:46–50). Indeed, centralized active speaker management was well-known prior to the '858 patent. (*See, e.g.*, Appx. N (Kumar), 4:21–36, 8:23–37 (describing the use of an MCU to control which terminals are actively multicasting audio); Appx. O (O'Malley), 1:65–2:19 (describing the use of a centralized audio mixer to determine which participants are

speaking and to remove a speaking participant's audio from the summed audio signal that participant receives).)

Oran further suggests this combination, mentioning that the number of entries in its data array (M) is limited by the size of memory at the client device. (*See* Appx. H (Oran), 4:32–35 ("Referring to FIG. 4, for any given audio session, there is some number of potential speakers 26. Call this value S. The receiver 24 has enough memory 38 to keep state information in data array 42 for some subset of these speakers 26. Call this value M.").) Based on these suggestions, a POSITA would have been motivated to perform active speaker management on a system having more memory so that state information can be tracked for more participants in a conference call. (Appx. C (Bress Decl.), ¶¶ 59–61.) A POSITA would recognize that this combination permits conference calls with a greater number of participants to be handled by the system. (Appx. C (Bress Decl.), ¶¶60–61; *see, e.g.*, Appx. H (Oran), 1:27–32 ("For example, an interactive conference call conducted for a seminar might include hundreds of callers. Individual receivers do not have the processing resources to even track the state information for every caller at the seminar."), 2:9–13 ("Depending on available processing resources, the speaker status, LRU time, and Talkspurt Packet Count, speaker entries are stored, discarded or changed in the data array and audio packets from speakers are either stored or discarded in memory."), 4:35–37 ("For simple three-way calling and

small audio conferences it is likely that M>S, but for larger conferences it is possible that S>>M.”).)

Further, a POSITA would have been motivated to use a system with enhanced processing resources, such as a centralized server, to generate state information for call participants. (Appx. C (Bress Decl.), ¶61.) Oran stresses that “retaining status information and decoding only a single audio stream is a process intensive operation. For example, over 15 Million Instructions Per Second (MIPs) is required to process a G.729 compressed voice stream.” (Appx. H (Oran), 3:31–33.) Indeed, Oran suggests the use of a more powerful system stating that “[i]f processing and memory were not a concern, state information could be kept at each receiver 24A-24D for all potential speakers 26A-26D, and for all those speakers who might be talking at any given time.” (*See* Appx. H (Oran), 3:26–30.)

Oran mentions several traffic and network factors to consider when implementing a centralized MCU. A POSITA would not be deterred from the combination based on any of these considerations. First, Oran states that “extra overhead is incurred by first sending all packets to the MCU 14 rather than simply multicasting the packets directly to each packet gateway 18.” (Appx. H (Oran), 2:58–60.) A POSITA would recognize that unicasting to, and multicasting from, the centralized MCU still reduces overhead of either unicasting from each participant to every other participant or unicasting a stream from the MCU to every

participant. (Appx. C (Bress Decl.), ¶54.) In addition, a POSITA would recognize that Oran's method of processing only packets for active speakers reduces the number of packets required to be multicast from the MCU. (Appx. C (Bress Decl.), ¶54.) Thus, a POSITA would recognize the benefits of above-referenced centralization and understand that any extra overhead is minimal (or negligible) and that bandwidth may actually be improved in the combined system. (Appx. C (Bress Decl.), ¶54.)

Oran notes that "MCU 18 uses a brut force method to mix of all speakers together and then send out the mixed signal to all packet gateways 18 in the conference call." (Appx. C (Bress Decl.), ¶55; *see* Appx. H (Oran), 2:61–63.) Pearce teaches that unmixed audio packets are sent to multicast receivers (receivers that can mix the audio), removing the need to mix centrally for all clients. (Appx. C (Bress Decl.), ¶55.) As taught by Pearce, mixed audio packets are sent by the multicast intermediaries to unicast receivers. (Appx. C (Bress Decl.), ¶55.) Additionally, as taught by Oran, only audio from active speakers is mixed and therefore the resources used for mixing are significantly less than implied by Oran for a prior-art MCU. (Appx. C (Bress Decl.), ¶55.)

Oran states that "the MCU 18 represents a single point of failure in the network." (Appx. 56 (Bress Decl.), ¶X; *see* Appx. H (Oran), 2:66–67.) As Oran notes in its Background section, MCUs or centralized "audio bridges" were used to

155853739.8
155853739.12

handle conference calls in circuit-switched systems for many years prior to Oran. (*See* Appx. H (Oran), 1:16–20.) Techniques for providing back-ups and system redundancy were common in such networks and would have been obvious to a POSITA. (Appx. C (Bress Decl.), ¶56.)

A POSITA would further understand that the combined system achieves Pearce's purpose of "enabling multicast telecommunications" to minimize network traffic because the MCU is a multicast source and the IP telephony devices and multicast intermediary are multicast receivers. (Appx. C (Bress Decl.), ¶49; *see* Appx G (Pearce), 1:40–52.) Additionally, the combined system achieves Pearce's goal of allowing "multicast telephony devices to communicate with each other using multicast streaming while still allowing unicast telephony devices to participate." (Appx. C (Bress Decl.), ¶49; *see* Appx G (Pearce), 1:54–56.)

2. The combination of Pearce and Oran renders independent claim 1 obvious.

a. The combination of Pearce and Oran discloses the preamble.

To the extent the preamble is limiting, both Pearce and Oran disclose a "*method of providing audio conferencing for a plurality of clients using varying equipment and protocols*" [1P]. (Appx. C (Bress Decl.), ¶¶64–68.) In Pearce, "a method is provided for enabling a multicast telecommunication session." (Appx. G (Pearce), 1:58–60.) Pearce's "method includes receiving multicast media streaming sent to a multicast group address at a multicast intermediary" and "communicating

the media streaming to a unicast telephony device to enable the unicast telephony device to participate in a multicast telecommunication session." (Appx. G (Pearce), 1:60–65.)

A POSITA would understand that Pearce's "telecommunication session" is an "*audio conference.*" (Appx. C (Bress Decl.), ¶65.) Pearce describes calls "made between an IP telephony device located on a LAN 20 and another IP telephony device located on another LAN 20, across WAN 30, or on Internet 40." (Appx. G (Pearce), 5:50–53.) Pearce explains that "[i]n the case of an IP telephone, as the user speaks into the handset, the codec [coder/decoder] converts the analog voice signals into digital data" which is "then encapsulated into IP packets so that it can be transmitted over LAN 20a." (Appx. G (Pearce), 4:51–55.) Pearce further describes calls between IP and non-IP devices. In these calls, incoming voice transmissions from non-IP devices are received at a gateway and converted "into the digital format used by the LAN." (Appx G (Pearce), 5:28–31.)

Pearce's method and system supports "*a plurality of clients using varying equipment and protocols*." (Appx. C (Bress Decl.), ¶¶66–67.) An exemplary embodiment of Pearce's network, illustrated in Figure 1 below, includes IP telephony devices 22–24 which "have the capability of encapsulating a user's voice (or other media inputs) into IP packets." (Appx. G (Pearce), 3:52–55.) Pearce teaches that IP telephony devices "may include telephones, fax machines,

155853739.8
155853739.12

computers running telephony software (such as MICROSOFT NETMEETING), analog or digital gateways, or any other device capable of performing telephony functions using an IP network." (Appx G (Pearce), 3:55–60.) Pearce's network also includes "non-IP telephony devices 54, 68 [e.g., PSTN telephony device][3] that are connected to PBX 50 or PSTN 60." (Appx. G (Pearce), 5:9–11.) Calls to and from these non-IP telephony devices "are made through a gateway 52, 64." (Appx. G (Pearce), 5:11–12.) Because "the digital format for voice transmissions over an IP network is often different than the format used on the digital trunks of PSTN 60, gateway 64 provides a conversion between these different digital formats, referred to as transcoding." (Appx. G (Pearce), 5:20–24.)

Oran also discloses a "*method of providing audio conferencing for a plurality of clients using varying equipment and protocols.*" (Appx. C (Bress Decl.), ¶68.) Oran is directed to a system that "adaptively select[s] which audio packets to mix together as an audio signal depending on the active status of the multiple speakers." (Appx. H (Oran), 1:8–12.) Oran's MCU "takes the responsibility of listening to all speakers that may talk via telephones 16." (Appx. H (Oran), 2:47–49.) Oran repeatedly refers to these multiple speaker sessions as

[3] Herein, language included in brackets is added to quotations by Requestor for clarification.

conferences. (*See* Appx. H (Oran), 2:63, 3:23–25 ("For example, the gateways 24A-D and telephones 16 could be replaced with personal computers (PC's) running a **conferencing application**."), 3:58–60 ("The receiver in step 34 retains information on selected speakers that appear to be actively participating in the **telephone conference**.").) Because Oran supports conventional PSTN telephones, gateways, as well as PC's "running a conferencing application", Oran's network supports "*varying equipment and protocols*." (Appx. C (Bress Decl.), ¶68.)

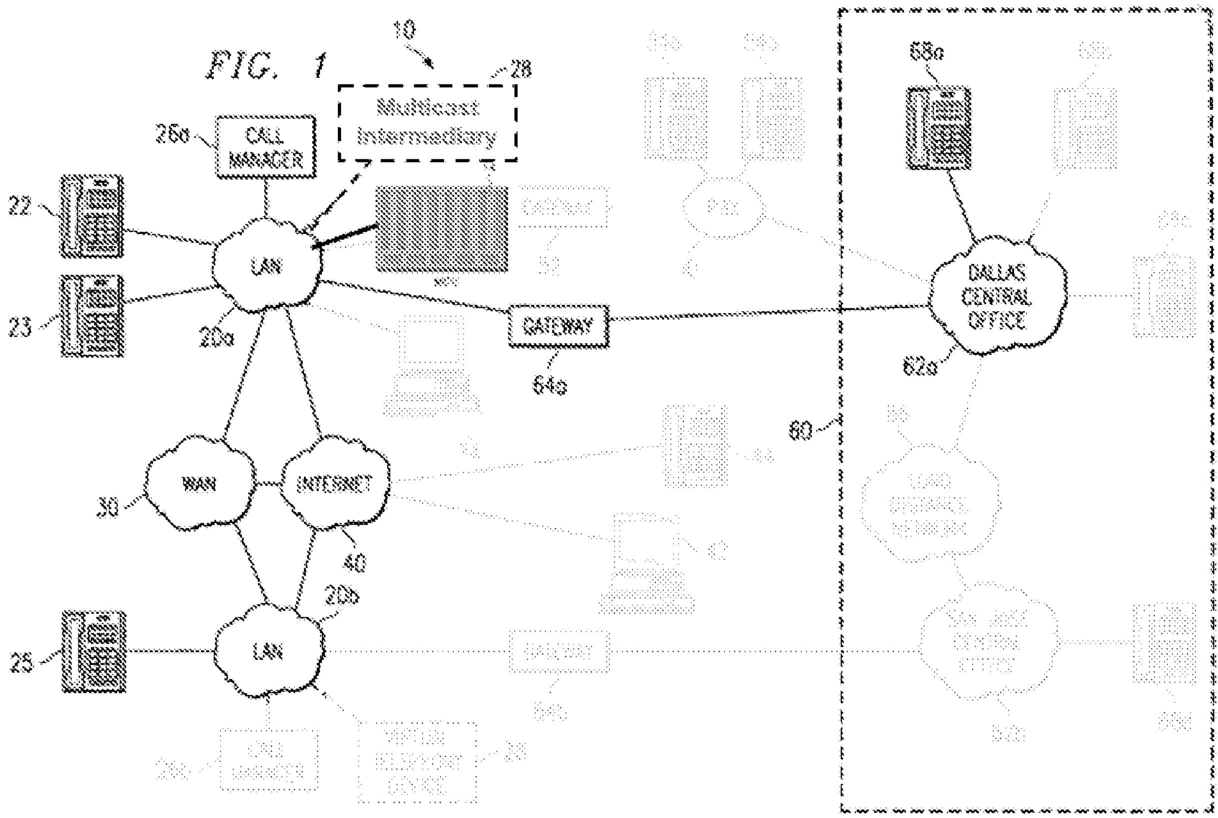
Accordingly, the combination of Pearce and Oran discloses a "*method of providing audio conferencing for a plurality of clients using varying equipment and protocols*." (Appx. C (Bress Decl.), ¶¶64–68.)

b. The combination of Pearce and Oran discloses "receiving an audio packet from each of the plurality of clients" [1.1].

The combination of Pearce and Oran discloses "*(1) receiving an audio packet from each of the plurality of clients*" [1.1]. (Appx C (Bress Decl.), ¶¶69–72.) Pearce discloses an exemplary telecommunications session between a PSTN telephony device 68a (through gateway 64a) and one or more IP telephony devices 22, 23, 25. (*See* Appx. G (Pearce), 10:55–61.) An exemplary network for this telecommunication session is illustrated in Bress Figure B which annotates Pearce Figure 1 below to gray out elements not involved in the session, add the multicast intermediary 28 in place of the equivalent "virtual telephony device 28" (as taught

by Pearce)[4], and overlay Oran's MCU (shaded red) into the combined system. (Appx. C (Bress Decl.), ¶70.)

[4] "Virtual telephony devices may be logically inserted between two or more IP telephony devices **to act as an intermediary between the two telephone devices**." (Appx. G (Pearce), 6:54–56.) Pearce explains that "[o]nce such a relationship is set up, signaling and media streaming that passes through the virtual telephony device may then be modified through address translation or data stream manipulation for various reasons before they are sent on to the destination device." (Appx. G (Pearce), 6:56–61.)



Bress Figure B

Pearce explains that call manager 26a "establishes a multicast group having a multicast group address 100" and "instructs telephony devices 22, 23 and 25 to initiate outgoing media streaming 104 to multicast address 100." (Appx. G (Pearce), 11:4–6.) In the combined system, as discussed in the motivation to combine section, multicast address 100 is the multicast address for the multicast clients in the telecommunication session. (Appx. C (Bress Decl.), ¶71.) Because gateway 64a (associated with PSTN telephony device) is a unicast device, call manager 26a "generates a multicast intermediary" and "associates a logical port of multicast intermediary 28 with each of telephony device 64a and **multicast**

address 100." (Appx. G (Pearce), 11:14–18.) Specifically, gateway 64a receives "analog or digital circuit-switched data transmitted by PSTN 60" from PSTN telephony device 68 and converts it "to packetized data transmitted by LAN 20." (Appx. G (Pearce), 5:14–16.) Gateway 64a sends its packet "to the logical port of multicast intermediary 28" as instructed by call manager 26a. (Appx. G (Pearce), 11:22–25.)

In the combination of Pearce and Oran, the client devices (e.g., IP telephony devices and multicast intermediaries) send their respective IP audio packets to the MCU as unicast packets. (Appx. C (Bress Decl.), ¶72.) In Pearce, audio data for a telecommunication session is "encapsulated into IP packets so that it can be transmitted over LAN 20a." (Appx. G (Pearce), 4:51–55; *see also*, 5:14–16.) Therefore, the multicast intermediary/gateway 64a and IP telephony devices 22, 23, and 25 (the recited "*plurality of clients*") each transmits "*an audio packet*" to the MCU. (Appx. C (Bress Decl.), ¶72.) Oran explains that the centralized MCU "takes the responsibility of listening to all speakers that may talk" and in Oran's Figure 1 shows data being communicated across a "packet network." (Appx. H (Oran), 2:48–49; Figure 1, *see also*, 2:54–56 (describing the packet gateway as converting between audio packets and audio signals).)

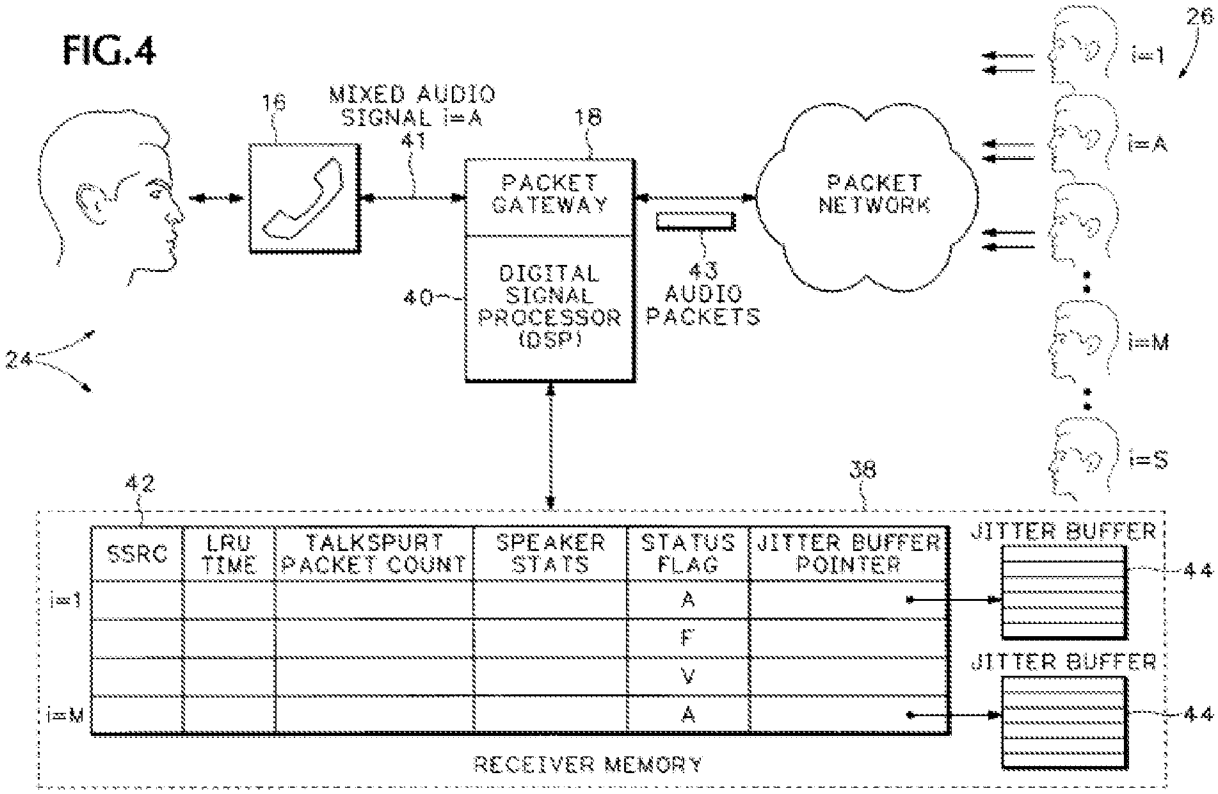
Accordingly, the combination of Pearce and Oran discloses "*receiving an audio packet from each of the plurality of clients*" [1.1]. (Appx. C (Bress Decl.), ¶¶69–72.)

c. The combination of Pearce and Oran discloses "(2) determining which of the plurality of clients is an active speaker …" [1.2].

The combination of Pearce and Oran discloses "*(2) determining which of the plurality of clients is an active speaker and forming an active speakers list*" [1.2]. (Appx. C (Bress Decl.), ¶¶73–76.) As discussed in the Section V.A.1 (Overview of the Combination), it would have been obvious to a POSITA to integrate Oran's active speaker management into the centralized MCU in the combination of Pearce and Oran.

In Oran's active speaker management process, the receiver (the MCU in the combined system) "receives audio packets and identifies the speakers 26A-26D associated with the audio packets." (Appx. H (Oran), 3:40–42.) The receiver then "decides which speakers are currently 'active' and warrant keeping information about." (Appx. H (Oran), 3:44–45.) As illustrated in Oran Figure 4 (reproduced below), state information is generated and managed using a digital signal processor (DSP). (Appx. H (Oran), Figure 4, *see also*, 5:47–7:35.) Specifically, "data array 42 keeps information about the M speakers." (Appx. H (Oran), 5:8–9.)

FIG.4



Each row in the "data array 42 represents a speaker entry containing the state information for one of the M speakers." (Appx. H (Oran), 5:9–10.) SSRC "is used to match speaker entries in the data array 42 with audio packets 43." (Appx. H (Oran), 5:11–12.) Speaker Stats "contain a variety of information about a speaker 26 used for sizing and operating a jitter buffer 44 when the speaker 26 is active." (Appx. H (Oran), 5:28–30.) The Status Flag "V" "is used to identify either an inactive speaker or a speaker who is being ignored in favor of another speaker who is active" and the Status Flag "F" "identifies a speaker free to be used or reused by another speaker 26." (Appx. H (Oran), 5:40–46.) Thus, speaker entries having a Status Flag "A" are "*active speakers*" and data array 42 includes an "*active*

155853739.8
155853739.12

speakers list." (Appx. C (Bress Decl.), ¶75.) Should Patent Owner contend that the "*active speakers list*" must include only active speaker entries and no other entries, it would have been obvious to a POSITA to form a data array with only active speakers and/or form a separate data array with data for non-active speakers. (Appx. C (Bress Decl.), ¶75.) Such a modification is a mere implementation detail well within the skill of a POSITA because the modification merely removes entries from a list. (Appx. C (Bress Decl.), ¶75.) A list is merely a "multi-element data structure that has a linear (first, second, third, …) organization but that allows elements to be added or removed in any order." (Appx. E (Microsoft Dictionary), 270.) A list is one of the most natural and oldest forms of organizing data. (Appx. C (Bress Decl.), ¶75.) And, even prior to the '858 patent, lists were one of the very first data structures taught in programming. (Appx. C (Bress Decl.), ¶75.) Therefore, creating and modifying a list would have been well-known and understood to a POSITA. (Appx. C (Bress Decl.), ¶75.)

Accordingly, the combination of Pearce and Oran discloses or at least renders obvious "*(2) determining which of the plurality of clients is an active speaker and forming an active speakers list*" [1.2]. (Appx C (Bress Decl.), ¶¶73–76.)

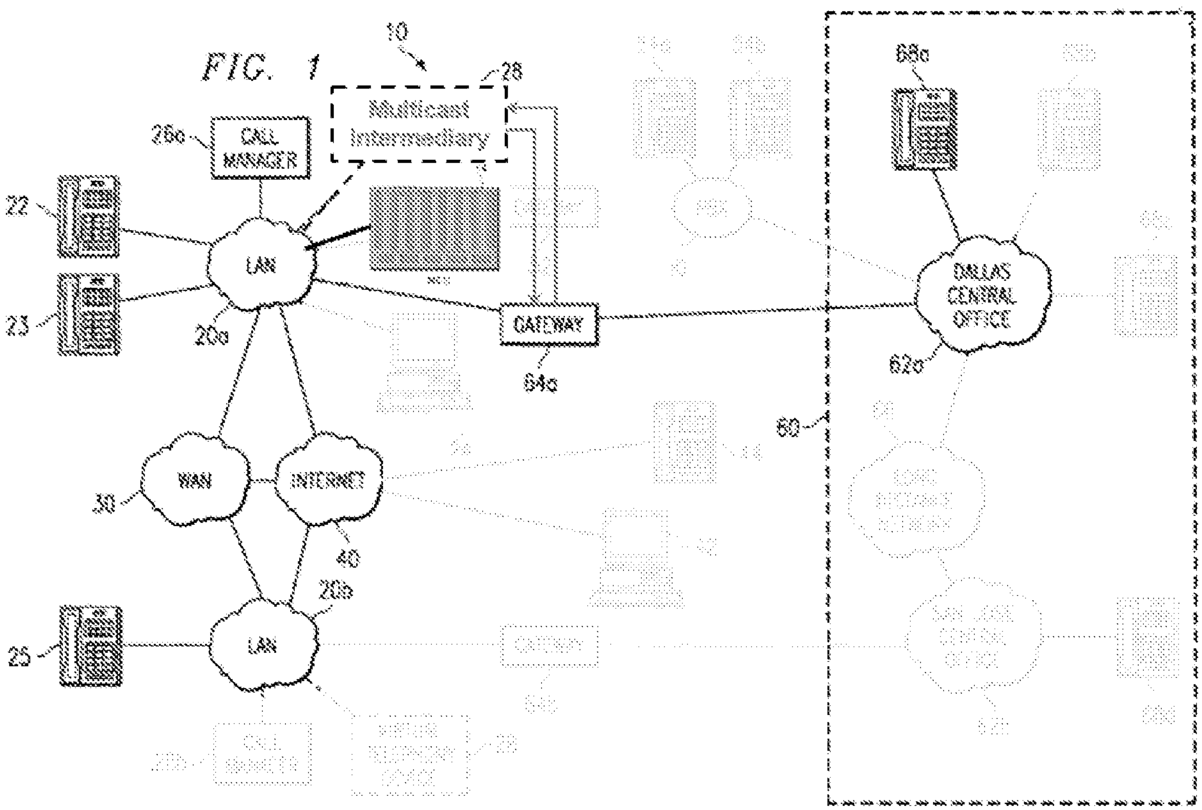
d. The combination of Pearce and Oran discloses "(3) determining that a first subset of the plurality of clients has the capability to mix …" [1.3] and "(4) determining that a

second subset of the plurality of clients does not have the capability to mix …" [1.4].

The combination of Pearce and Oran discloses "*(3) determining that a first subset of the plurality of clients has the capability to mix multiple audio streams*" [1.3] and "*(4) determining that a second subset of the plurality of clients does not have the capability to mix multiple audio streams*" [1.4]. (Appx. C (Bress Decl.), ¶¶77–83.)

Pearce describes an exemplary telephone session (conference call) "between multicast telephony devices 22, 23, 25 and a unicast telephony device 64a using a multicast intermediary 28." (Appx. G (Pearce), 10:43–45; 10:12-15 (explaining that gateways are unicast devices).) The session is illustrated in the exemplary network diagram of Bress Figure C which further annotates Bress Figure B to illustrate the connection between gateway 64a and multicast intermediary 28. (Appx. C (Bress Decl.), ¶78.) "Standing alone, unicast telephony devices are not able to participate in a multicast telecommunication session because unicast telephony devices are not capable of monitoring a multicast group address to determine if any messages are being sent to the multicast group." (Appx. G (Pearce), 10:16–21.) As shown in Bress Figure C below, multicast intermediary 28 "is inserted into a telecommunication session on behalf of" unicast telephony device 64a so that unicast telephony device (gateway) 64a "can effectively

participate in a multicast telecommunication session." (Appx. G (Pearce), 10:31–34.)



Bress Figure C

To establish the telecommunication session (e.g., a conference call), telephony device 64a (gateway) "sends a call initiation request using TCP signaling 102 (or any other appropriate type of signaling) to call manager 26a indicating that PSTN telephony device 68a requests a telecommunication session (e.g., a conference call) with telephony devices 22, 23 and 25." (Appx. G (Pearce), 10:55–61.) If telephony devices 22, 23 and 25 can participate, "call manager 26a

155853739.8
155853739.12

initiates the telecommunication session between the telephony devices." (Appx. G (Pearce), 10:64–67.)

A call manager "is an application that controls call processing, routing, telephone features and options (such as call hold, call transfer and caller ID), **device configuration**, and other telephony functions and parameters within communication network 10." (Appx. G (Pearce), 4:9–13.) A single call manager 26a "can control all of the IP telephony devices on LAN 20a, and it may also control IP telephony devices located across WAN 30." (Appx. G (Pearce), 4:13–15.) That is, in the exemplary conference call of Bress Figure C, call manager 26a controls IP telephony devices 22, 23, and 25.

"From registration information previously sent by telephony devices 22, 23 and 25 to call manager 26a, call manager 26a determines that telephony devices 22, 23 and 25 **support multicast communication**." (Appx. G (Pearce), 11:1–4.) Similarly, "call manager 26a determines from either registration information or the call initiation request that telephony device 64a is a **unicast telephony device**." (Appx. G (Pearce), 11:12–14.) In the exemplary call described in Pearce and illustrated in Bress Figure C, call manager determines that IP telephony devices 22, 23, and 25 are multicast devices and that gateway 64a is a unicast device. Thus, Pearce's call manager "*determine[es] that a first subset of the plurality of clients*" supports multicast communication and "*determine[es] that a second subset of the*

plurality of clients does not" support multicast (i.e., are "unicast devices"). (Appx. C (Bress Decl.), ¶81.)

Pearce teaches that a unicast device (i.e., a device that is not multicast-capable) is "not able to determine the original source of the packets included in media streaming 108 (telephony device 64a believes all packets originated at multicast intermediary 28), and thus telephony device 64a cannot properly sort and sequence the incoming data." (Appx. G (Pearce), 11:46–51.) In the combined system, the lack of ability to properly sort and sequence prevents a unicast device from being able to mix multiple audio streams in a conference call. (Appx. C (Bress Decl.), ¶82.)

Accordingly, by affirmatively determining whether a device is a multicast device or a unicast device (not multicast-capable), the combination of Pearce and Oran discloses "*(3) determining that a first subset of the plurality of clients has the capability to mix multiple audio streams*" [1.3] and "*(4) determining that a second subset of the plurality of clients does not have the capability to mix multiple audio streams*" [1.4]. (Appx. C (Bress Decl.), ¶¶77–83.)

e. The combination of Pearce and Oran discloses or at least suggests "(5) multiplexing said packets of audio data received

from each client on said active speakers list into a multiplexed stream" [1.5].

The combination of Pearce and Oran discloses or at least suggests "*(5) multiplexing said packets of audio data received from each client on said active speakers list into a multiplexed stream*" [1.5]. (Appx. C (Bress Decl.), ¶¶85–88.)

The MCU in the combined system of Pearce and Oran identifies "*packets of audio data received from each client on said active speakers list*" to send to multicast capable devices (i.e., devices that have the capability to mix). (Appx. C (Bress Decl.), ¶87.) As discussed previously, Oran teaches that a receiver (the MCU in the combined system) "decides which speakers are currently 'active' and warrant keeping information about." (Appx. H (Oran), 3:43–44.) "Of the speakers who are identified 'active'," the receiver "decides which speakers to mix together to produce an output audio stream." (Appx. H (Oran), 3:61–63.)

In the combined system of Pearce and Oran, active speaker audio packets are not mixed in the MCU for multicast devices. (Appx. C (Bress Decl.), ¶87.) Instead, the audio packets are multicast in a single IP output stream. it's a multicast telephony devices receives the streaming at a multicast address which, in the combined system, is the multicast address assigned by the call manager for the multicast clients. (*See, e.g.,* Appx. G (Pearce), 9:61–65.) Each multicast "telephony device then mixes or sums the media streaming received from each of the other telephony devices to form a conference-like input (this may be referred to as a

client summing multicast conference call).” (Appx. G (Pearce), 9:65–10:2.) Therefore, in the combined system, the generated “output audio stream” is a sequence of IP audio packets for the identified active speakers. (Appx. C (Bress Decl.), ¶88.) This “output audio stream” of IP audio packets is provided to multicast capable devices which then perform the audio mixing. (Appx. C (Bress Decl.), ¶87.)

As discussed in Section II.D (Claim Construction), the parties in the co-pending district court action dispute the plain and ordinary meaning (and hence the broadest **reasonable** interpretation) of the term “*multiplexed stream*.” Patent Owner appears to contend that a “*multiplexed stream*” is nothing more than a sequence of audio packets. This understanding of the breadth of Patent Owner’s interpretation is supported by Patent Owner’s infringement contentions which appear to rely solely on transmission over TCP or UDP to show “*multiplexing said packets of audio data … into a multiplexed stream*”:

1[e] (5) multiplexing said packets of audio data received from each client on said active speakers list into a multiplexed stream;	Webex practices “multiplexing said packets of audio data received from each client on said active speakers list into a multiplexed stream” for at least the reasons described *supra* at 1[a]-[d]. Webex multiplexes audio packets received from attendees into a multiplexed stream. *See* Cisco, *What's New in Cisco Webex Analytics and Troubleshooting*, available at https://help.webex.com/en-us/n45okcs/What-s-New-in-Cisco-Webex-Analytics-and-Troubleshooting (indicating audio transport configuration changes which affect the multiplexing scheme for UDP vs. TCP data streams).

(Appx. Q (Patent Owner Infringement Contentions), p. 12)

A POSITA would understand that the “output audio stream” generated by the MCU for multicast devices in the combination of Pearce and Oran is a

"*multiplexed stream*" under Patent Owner's interpretation of the term. (Appx. C (Bress Decl.), ¶88.) The conferencing application in the combination of Pearce and Oran takes audio packet streams from many sources (e.g., IP telephony devices 22, 23, 25 and gateway 64a) and outputs packets from each of the active speaker sources in a single IP packet stream (i.e., one packet after the other). (Appx. C (Bress Decl.), ¶88.) The "output audio stream" in the combined system is therefore a stream of IP packets consisting of audio packets from each client that is an active speaker. (Appx. C (Bress Decl.), ¶¶87–88.) The "output audio stream" generated by the conferencing application therefore meets Patent Owner's broad interpretation of the term "*multiplexed stream*." (Appx. C (Bress Decl.), ¶88.)

Accordingly, the combination of Pearce and Oran discloses or at least suggests "*(5) multiplexing said packets of audio data received from each client on said active speakers list into a multiplexed stream*" [1.5] under Patent Owner's broad interpretation.[5] (Appx. C (Bress Decl.), ¶¶85–88.)

---

[5] Requestor demonstrates that the combination of Pearce, Oran, and either of the RTP multiplexing references (Hoshi or Rosenberg) discloses this limitation under Requestor's interpretation of the plain and ordinary meaning of "*multiplexed stream*."

f. The combination of Pearce and Oran discloses "(6) sending said multiplexed stream to each of said first subset of the plurality of clients" [1.6].

The combination of Pearce and Oran discloses "*(6) sending said multiplexed stream to each of said first subset of the plurality of clients*" [1.6]. (Appx. C (Bress Decl.), ¶¶90–91.) In the combination of Pearce and Oran, the MCU (acting as a multicast source) multicasts the "output audio stream" containing the IP audio packets for each speaker identified in the active speaker list to the multicast address set by the call manager for participating multicast clients. (Appx. C (Bress Decl.), ¶91; *see also* Section V.A.1.c (describing MCU acting as a multicasting source in the combined system).) To send IP multicast packets, the source device (the MCU in the combined system) "specifies a destination address that represents the multicast group" assigned by the call manager for the conference call. (Appx. G (Pearce), 9:39–41.) The multicast IP packets are transmitted to a multicast-enabled router which "forwards [the] multicast message to a particular network segment only when there are multicast receivers on that network segment." (Appx. G (Pearce), 9:25–30.)

As discussed in the prior sections, the "*multiplexed stream*" in the combination of Pearce and Oran is sent only to multicast clients which have the capability to mix the audio locally which include the recited "*first subset of the plurality of clients*" and the multicast intermediary via the multicast address set by

the call manager for multicast clients participating in the telecommunication session (e.g., the conference call). (Appx. C (Bress Decl.), ¶91.) The unicast device (gateway 64) does not receive the "*multiplexed stream*." (Appx. C (Bress Decl.), ¶91.)

The combination of Pearce and Oran therefore discloses "*(6) sending said multiplexed stream to each of said first subset of the plurality of clients*" [1.6]. (Appx. C (Bress Decl.), ¶¶90–91.)

g. The combination of Pearce and Oran discloses "(7) mixing said packets of audio data received from each client on said active speakers list into one combined packet" [1.7].

The combination of Pearce and Oran discloses "*(7) mixing said packets of audio data received from each client on said active speakers list into one combined packet*" [1.7]. (Appx. C (Bress Decl.), ¶93–95.)

As discussed above, a unicast telephony device "is not able to determine the original source of the packets included in media streaming 108 (telephony device 64a believes all packets originated at multicast intermediary 28), and thus telephony device 64a cannot properly sort and sequence the incoming data." (Appx. G (Pearce), 11:46–51). Because of this limitation, the unicast device cannot locally mix the audio data of the active speakers. (Appx. C (Bress Decl.), ¶94.) Pearce therefore discloses that the "multicast intermediary will typically sort **or mix media streaming 104** received from telephony devices 22, 23 and 25."

(Appx. G (Pearce), 11:43–46.) Oran describes mixing as a function performed by a network-based component: "MCU 18 uses a brute force method to mix of [sic] all speakers together and then send out the mixed signal to all packet gateways in the conference call." (Appx. H (Oran), 2:61–63.) Thus, the combination of Pearce and Oran discloses "*mixing said packets of audio data*" for any unicast clients in the conference call such as gateway 64a. (Appx. C (Bress Decl.), ¶94.)

The mixing performed in the combined system is of the "*packets of audio data received from each client on said active speakers list*." (Appx. C (Bress Decl.), ¶95.) As discussed for limitation [1.5], Oran discloses that "[o]f the speakers who are identified 'active'," the receiver "decides which speakers to mix together to produce an output audio stream." (Appx. H (Oran), 3:61–63.) Accordingly, in the combined system, only the audio packets from speakers identified as active are received by the multicast intermediary and mixed together. (Appx. C (Bress Decl.), ¶95.) Like the "*multiplexed packet*", the mixed audio is transmitted in one or more IP packets to a router. (Appx. C (Bress Decl.), ¶95 (explaining the packetizing interval used for various codecs including 20 ms for GSM, G.729, and G.711 and 30 ms for G.723).) Each packet contains all or a portion of the mixed audio and therefore is a "*combined packet*." (Appx. C (Bress Decl.), ¶95.)

The combination of Pearce and Oran therefore discloses "*(7) mixing said packets of audio data received from each client on said active speakers list into one combined packet*" [1.7]. (Appx. C (Bress Decl.), ¶¶93–95.)

h. The combination of Pearce and Oran discloses "(8) sending said combined packet to each of said second subset of the plurality of clients" [1.8].

The combination of Pearce and Oran discloses "*(8) sending said combined packet to each of said second subset of the plurality of clients*" [1.8]. (Appx. C (Bress Decl.), ¶¶97–98.)

In the combination of Pearce and Oran, the multicast intermediary sends the mixed "*packets of audio data received from each client on said active speakers list*" to the unicast devices (e.g., telephony device 64a), which lack the capability to mix audio in multicast telecommunications sessions in the combined system (the recited "*second subset of the plurality of clients*"). (*See* Section V.A.2.d ([1.2])); *see also* Appx. C (Bress Decl.), ¶98.) (Specifically, "the multicast intermediary 28 **forwards media streaming 108** that it receives from multicast address 100 to telephony device 64a as media streaming 108 using the communication module… However, before the media streaming 108 is forwarded, multicast intermediary will typically sort or **mix media streaming 104** received from telephony devices 22, 23 and 25."(Appx. G (Pearce), 11:37–46.)

Accordingly, the combination of Pearce and Oran discloses "*(8) sending said combined packet to each of said second subset of the plurality of clients*" [1.8]. (Appx. C (Bress Decl.), ¶¶97–98.)

i. The combination of Pearce and Oran discloses "whereby said plurality of clients can simultaneously participate in a single audio conference application" [1.9].

The limitation "*whereby said plurality of clients can simultaneously participate in single audio conference application*" should not be afforded any patentable weight because it merely expresses the intended result of the process steps of claim. *See Hoffer v. Microsoft*, 405 F.3d 1326, 1329 (Fed. Cir. 2005) (noting that a "whereby clause in a method claim is not given weight when it simply expresses the intended result of a process step positively recited."). Regardless, the combination of Pearce and Oran discloses this claim element.

Pearce explains that "[t]hrough the use of multicast intermediary 28", a unicast telephony device 64a "may participate in a multicast telecommunication session in which it would not otherwise be capable of participating." (Appx. G (Pearce), 11:63–66.) Indeed, one of the stated "technical advantages" of Pearce is that its system and method "allow unicast telephony devices to effectively participate in a telecommunication session with multicast telephony devices." (Appx. G (Pearce), 2:11–14.) Thus, in the combined system, the "*plurality of*

clients can simultaneously participate in a single audio conference application" [1.9]. (Appx. C (Bress Decl.), ¶¶100–01.)

3. The combination of Pearce and Oran renders independent claim 6 obvious.

a. The combination of Pearce and Oran discloses the preamble.

To the extent the preamble is limiting, both Pearce and Oran disclose a "*system for providing audio conferencing for a plurality of clients*" [6P]. (Appx. C (Bress Decl.), ¶¶64–68.) In Pearce, "a system … for enabling multicast telecommunications is provided." (Appx. G (Pearce), 1:48–50.) In Pearce's system, "a communication network is provided that includes a unicast telephony device, and a plurality of multicast telephony devices operable to receive multicast media streaming transmitted to a multicast group address." (Appx. G (Pearce), 1:66–2:3.) The network "further includes a multicast intermediary operable to receive multicast media streaming sent to multicast group address" and to "communicate the media streaming to the unicast telephony device to enable the unicast telephony device to participate in the multicast communication with the multicast telephony devices." (Appx. G (Pearce), 2:3–10.) As explained in Section V.A.2.a, a POSITA would understand that Pearce's "multicast telecommunications" is "*audio conferencing*." (Appx. C (Bress Decl.), ¶65.)

Pearce's method and system supports "*a plurality of clients*." (Appx. C (Bress Decl.), ¶¶66–67.) As explained above, Pearce's network includes IP

telephony devices 22–24 which "have the capability of encapsulating a user's voice (or other media inputs) into IP packets" (Appx. G (Pearce), 3:52–55.), as well as "non-IP telephony devices 54, 68 that are connected to PBX 50 or PSTN 60" and make calls "through a gateway 52, 64." (Appx. G (Pearce), 5:9–12.)

Oran is similarly directed to a system that "adaptively select[s] which audio packets to mix together as an audio signal depending on the active status of the multiple speakers." (Appx. H (Oran), 1:8–12.) In the prior-art embodiment, Oran's MCU "takes the responsibility of listening to all speakers that may talk via telephones 16." (Appx. H (Oran), 2:47–49.) Oran repeatedly refers to these multiple speaker sessions as conferences. (*See* Appx. H (Oran), 2:63, 3:23–25, 3:58–60.)

Accordingly, the combination of Pearce and Oran discloses a "*system for providing audio conferencing for a plurality of clients*" [6P]. (Appx. C (Bress Decl.), ¶¶64–68.)

b. The combination of Pearce and Oran discloses or at least suggests "a receiver capable of receiving an audio packet from each of the plurality of clients" [6.1].

The combination of Pearce and Oran discloses or at least suggests "*a receiver capable of receiving an audio packet from each of the plurality of clients*" [6.1]. (Appx. C (Bress Decl.), ¶¶69–72.) For the reasons explained above with respect to limitation [1.1] of claim 1, the MCU in the combination of Pearce and

Oran discloses "*receiving an audio packet from each of the plurality of clients*." (*See supra* Section V.A.2.b.)

A POSITA would understand that, although not explicitly disclosed, the MCU includes "*a receiver capable of*" the disclosed "*receiving*." The term "*receiver*" is not defined in the '858 patent specification and thus must be afforded its broadest reasonable interpretation—i.e., any component capable of receiving one or more audio packets. As explained above in Section V.A.2.b ([1.1]), in the combination of Pearce and Oran, IP telephony devices and multicast intermediaries **send their respective IP audio packets to the MCU**. (Appx. C (Bress Decl.), ¶72.) Thus, the MCU includes a component capable of "*receiving*" packets. (Appx. C (Bress Decl.), ¶72.)

A POSITA would have understood that the MCU is implemented (or would have been motivated to implement the MCU) as a hardware device such a server that includes well-known components such as network interfaces and associated software for transmission and reception of packets over a network. (Appx. C (Bress Decl.), ¶72; *see also*, Appx. K (Robert), Abstract, Figures 2–3; Appx. M (Botzko), 3:65–4:7, 4:34–36 (describing RTP transport circuit for transmitting and receiving packets); Appx. O (O'Malley), Figure 2 (illustrating network interface cards).)

Accordingly, the combination of Pearce and Oran discloses or at least suggests "*a receiver* [e.g., network interface] *capable of receiving an audio packet from each of the plurality of clients*" [6.1]. (Appx. C (Bress Decl.), ¶¶69–72.)

c. The combination of Pearce and Oran discloses "means for maintaining a list of each of the plurality of clients that is an active speaker" [6.2].

The combination of Pearce and Oran discloses a "*means for maintaining a list of each of the plurality of clients that is an active speaker*" [6.2]. (Appx. C (Bress Decl.), ¶73–76.) As discussed in the Claim Construction section, the parties in the co-pending district court action agree that the function of this limitation is "maintaining a list of each of the plurality of the clients that is an active speaker" and the structure is "control logic, like that of control flow 300 in Figure 3, executed by the computer system 400 in Figure 4 as well as equivalents thereof." (*See supra* Section II.D.3.)

In the combined system, the centralized MCU performs the functions of "receiv[ing] audio packets and identif[ying] the speakers 26A–26D associated with the audio packets," "decid[ing] which speakers are currently 'active,'" and maintain[ing] state information for the speakers in data array 42. (Appx. H (Oran), 3:40–45, 5:8–10). Thus, Oran discloses the function of "*maintaining a list of each of the plurality of clients that is an active speaker*." (Appx. C (Bress Decl.), ¶76.)

155853739.8
155853739.12

Oran discloses that its "**processor** selects which audio packets and what speaker information to retain in memory" and "determines which of the selected audio packets to store in memory and mix together to produce an audio output signal." (Appx. H (Oran), Abstract.) Oran discloses that its active speaker management "can be employed in a pure-software or pure-hardware environment." (Appx. H (Oran), 4:49-51.) And Oran claims a "computer readable storage medium containing **software** for managing multiple speakers in a packet network" including steps related to speaker management. (Appx. H (Oran), claims 33-41.) A POSITA would therefore understand that the speaker management process of Oran is performed by the processor through the execution of control logic. (Appx. C (Bress Decl.), ¶76.)

Accordingly, the combination of Pearce and Oran discloses or at least renders obvious "*means for maintaining a list of each of the plurality of clients that is an active speaker*" [6.2]. (Appx. C (Bress Decl.), ¶¶73–76.)

d. The combination of Pearce and Oran discloses or at least suggests "means for storing information indicative of whether each of the plurality of clients has the capability to mix multiple audio streams" [6.3].

The combination of Pearce and Oran discloses or at least suggests "*means for storing information indicative of whether each of the plurality of clients has the capability to mix multiple audio streams*" [6.3]. (Appx. C (Bress Decl.), ¶¶77–84.) As discussed in the Claim Construction section, the parties in the co-pending

district court action agree that this limitation requires the function of "storing information indicative of whether each of the plurality of clients has the capability to mix multiple audio streams" to be performed by the structure of a "main memory 408 and processor 404 in Figure 4 as well as equivalents thereof." (*See supra* Section II.D.3.)

As discussed with respect to claim 1, call manager 26a "is an application that controls call processing, routing, telephone features and options (such as call hold, call transfer and caller ID), device configuration, and other telephony functions and parameters within communication network 10." (Appx. G (Pearce), 4:9–13; *see supra* Section V.A.2.d.) Pearce explains that to establish a telecommunication session, each telephony device sends a registration request to the call manager. (*See, e.g.*, Appx. G (Pearce), 7:7–23.) The registration message "typically comprises information about the telephony device such as the telephony device's IP and media access control (MAC) addresses, **the type and capabilities of the telephony device**, and the codec(s) used by the telephony device." (Appx. G (Pearce), 7:18–23.) Pearce further teaches that the call manager uses this registration "**previously** sent by telephony devices 22, 23, and 25" to determine "that telephony devices 22, 23 and 25 support multicast communication" and "that telephony device 64a is a unicast telephony device." (Appx. G (Pearce), 11:1–4, 11:12–14.) Although, Pearce does not explicitly state that this registration

information is "stored," a POSITA would understand that for the call manager to use "previously" provided information to determine an action to take, the information must be stored in a memory that is accessible by the call manager. (Appx. C (Bress Decl.), ¶84.) Accordingly, the combination of Pearce and Oran discloses or at least suggests performing the function of "*storing information indicative of whether each of the plurality of clients has the capability to mix multiple audio streams*."

Furthermore, Oran discloses the use of a memory to store information used by a platform. (Appx. H (Oran), Abstract (storing "information about the multiple speakers in the telephone call.").) Accordingly, the combination of Pearce and Oran teaching performing this function using a structure of a "main memory 408 and processor 404 in Figure 4 as well as equivalents thereof." A POSITA would have been motivated to implement the call manager on a platform (e.g., a server) having a processor and memory as taught by Oran. (Appx. C (Bress Decl.), ¶84.) Indeed, implementing an application such as a call manager on a computer having a processor and memory (e.g., server) would be a necessary, and well-understood, design choice for a POSITA. (Appx. C (Bress Decl.), ¶84.) Additionally, a POSITA would have been motivated to implement the call manager on the same hardware platform as the MCU. (Appx. C (Bress Decl.), ¶84.) A POSITA would have been motivated to implement these two network-based applications in a

single hardware platform to simplify maintenance and reduce costs. (Appx. C (Bress Decl.), ¶84.) For example, a POSITA would recognize that the call manager and MCU would share hardware components such as network interfaces, processors, and memory. (Appx. C (Bress Decl.), ¶84.)

The combination of Pearce and Oran therefore discloses or at least suggests "*means for storing information indicative of whether each of the plurality of clients has the capability to mix multiple audio streams*" [6.3]. (Appx. C (Bress Decl.), ¶¶77–84.)

e. The combination of Pearce and Oran discloses or at least suggests "a multiplexor capable of multiplexing said packets of audio data received from each client on said list of active speakers into a multiplexed stream" [6.4].

The combination of Pearce and Oran discloses or at least suggests "*a multiplexor capable of multiplexing said packets of audio data received from teach client on said list of active speakers into a multiplexed stream*" [6.4]. (Appx. C (Bress Decl.), ¶¶85–89.)

As explained in detail with respect to claim 1, a POSITA would understand that the "output audio stream" generated by the MCU in the combined system of Pearce and Oran is a "*multiplexed stream*" under Patent Owner's broad interpretation of the term. (*See supra* Section V.A.2.e.) Specifically, the MCU takes audio packet streams from many sources (e.g., IP telephony devices 22, 23, 25 and gateway 64a) and outputs packets from each of those sources in an "output

audio stream" to be provided to multicast capable devices which then perform the audio mixing. (Appx. C (Bress Decl.), ¶87.) The generated "output audio stream" is a sequence of IP audio packets for the identified active speakers, and thus satisfies Patent Owner's interpretation of a "*multiplexed stream*." (Appx. 88 (Bress Decl.), ¶X; *see also* Section V.A.2.e.)

A POSITA would understand that the MCU in the combination includes a component for performing the disclosed "*multiplexing*." (Appx. C (Bress Decl.), ¶89.) Indeed, the term "multiplexor" is not defined in the specification and thus must be afforded its broadest reasonable interpretation—i.e., any component capable of multiplexing packets of data. In the combined system, Pearce's IP telephony devices and multicast intermediary send their respective IP audio packets to the MCU. (*See supra* Section V.A.3.b.) The MCU, in turn, generates an output audio stream which is a sequence of IP audio packets. Thus, in the combination of Pearce and Oran, the MCU includes the component producing the single multiplexed output stream from multiple input streams and therefore is "*capable of multiplexing said packets of audio data received from each client on said list of active speakers into a multiplexed stream*" under Patent Owner's interpretation.

Accordingly, the combination of Pearce and Oran discloses or at least suggests "*a multiplexor capable of multiplexing said packets of audio data*

received from teach client on said list of active speakers into a multiplexed stream" [6.4] under Patent Owner's broad interpretation.[6]

f. The combination of Pearce and Oran discloses "a mixer capable of mixing said packets of audio data received from each client on said list of active speakers into one combined packet" [6.5].

The combination of Pearce and Oran discloses "*a mixer capable of mixing said packets of audio data received from each client on said list of active speakers into one combined packet*" [6.5]. (Appx. C (Bress Decl.), ¶¶93–96.) As explained in detail above with respect to claim 1, the combination of Pearce and Oran performs the function of "*mixing said packets of audio data received from each client on said list of active speakers into one combined packet*." (*See supra* Section V.A.2.g.)

A POSITA would understand that, in the combined system of Pearce and Oran, the multicast intermediary includes a "*mixer capable of*" the disclosed "*mixing*." (Appx. C (Bress Decl.), ¶96.) As explained above in Section V.A.2.h, the multicast intermediary sends the combined packet of audio data to unicast devices (e.g., telephony device 64a), which lack the capability to mix audio in

---

[6] Requestor demonstrates that the combination of Pearce, Oran and Hoshi or Rosenberg discloses this limitation under Requestor's interpretation of the plain and ordinary meaning. (*See infra* Section V.B.)

multicast telecommunications sessions. Before doing so, Pearce teaches that the "multicast intermediary will typically sort or **mix** media streaming 104 received from telephony devices 22, 23 and 25." (Appx. G (Pearce), 11:37–46; *see also* 11:46–55 ("Since telephony device 64a is not multicast-capable, it is not able to determine the original source of the packets included in media streaming 108… Therefore, multicast intermediary can either mix media streaming 104 received from telephony devices 22, 23, and 25…"), 12:19–21 ("multicast intermediary 28a forwards multicast streaming 106 to telephony device 64a (after any appropriate mixing or sorting)").) Oran explicitly describes a component that performs mixing. (*See* Appx. H (Oran), 4:57-61 ("The DSP 40 (or software implementation) works in conjunction with the packet gateways 18 and has the processing speed to actually decode and mix the audio from only a subset of speakers (A) from the M speakers it has state information about.").)

Accordingly, the combination of Pearce and Oran teaches "*a mixer capable of mixing said packets of audio data received from each client on said list of active speakers into one combined packet*" [6.5]. (Appx. C (Bress Decl.), ¶¶93–96.)

g. The combination of Pearce and Oran discloses "*a packet sender capable of sending …* " [6.6].

The combination of Pearce and Oran discloses "*a packet sender capable of sending, based on information in said means for storing, said multiplexed stream to each of the plurality of clients which have the capability to mix multiple audio*

streams" [6.6a]. (Appx. C (Bress Decl.), ¶¶90–92, 97–99.) As explained in detail above with respect to claim 1, the MCU in the combined system of Pearce and Oran (acting as a multicast source) multicasts the "output audio stream" containing the IP audio packets for each speaker identified in the active speaker list to multicast clients "*which have the capability to mix multiple audio streams*" locally (determined based on the recited "*information in said means for storing*"). (*See supra* Sections V.A.2.f & V.A.3.d.)

As discussed in Section V.A.2.f ([1.6]), in the combination, the MCU multicasts—i.e., sends—the "output audio stream" "*to each of the plurality of clients which have the capability to mix multiple audio streams*." (Appx. C (Bress Decl.), ¶91.) And as explained in Section V.A.2.h ([1.8]), the multicast intermediary sends the mixed "*packets of audio data received from each client on said active speakers list*" to the unicast devices "*which do not have the capability to mix multiple audio streams*." (*See also* Appx. 98 (Bress Decl.), ¶X (citing Appx. G (Pearce), 11:37–46).)

A POSITA would understand that both the MCU and the multicast intermediary in the combined system include a component capable the disclosed "*sending*." Indeed, the term "packet sender" is not defined in the specification and thus must be afforded its broadest reasonable interpretation—i.e., any component capable of sending packets of data. A POSITA would have understood that the

MCU and multicast intermediary are implemented (or would have been motivated to implement the MCU and multicast intermediary) as hardware devices such a server that includes well-known components such as network interfaces and associated software for transmission and reception of packets over a network. (Appx. 92 (Bress Decl.), ¶¶92, 99; *see also*, Appx. K (Robert), Abstract, Figures 2-3; Appx. M (Botzko), 3:65-4:7, 4:34-36 (describing RTP transport circuit for transmitting and receiving packets); Appx. O (O'Malley), Figure 2 (illustrating network interface cards).)

Accordingly, the combination of Pearce and Oran teaches "*a packet sender capable of sending, based on information in said means for storing, said multiplexed stream to each of the plurality of clients which have the capability to mix multiple audio streams, and capable of sending said combined packet to each of the plurality of clients which do not have the capability to mix multiple audio streams*" [6.6]. (Appx. C (Bress Decl.), ¶¶90–92, 97–99.)

h. The combination of Pearce and Oran discloses "whereby the plurality of clients can simultaneously participate in a single audio conference application" [6.7].

The limitation "*whereby the plurality of clients can simultaneously participate in single audio conference application*" should not be afforded any patentable weight because it "merely states the result of the limitations in the claim [and thus] adds nothing to the substance of the claim." *Lockheed Martin Corp. v.*

Space Sys./Loral, Inc., 324 F.3d 1308, 1319 (Fed. Cir. 2003) (discussing "whereby" clauses in the context of means-plus-function claims). Regardless, the combination of Pearce and Oran discloses this claim element.

As discussed above with respect to claim 1, Pearce explains that "[t]hrough the use of multicast intermediary 28," a unicast telephony device 64a "may participate in a multicast telecommunication session in which it would not otherwise be capable of participating." (Appx. G (Pearce), 11:63–66; *see also* Section V.A.2.i.) Indeed, one of the stated "technical advantages" of Pearce is that its system and method "allow unicast telephony devices to effectively participate in a telecommunication session with multicast telephony devices." (Appx. G (Pearce), 2:12–14.) Thus, in the combined system, "*the plurality of clients can simultaneously participate in a single audio conference application*" [6.7]. (Appx. C (Bress Decl.), ¶¶100–01.)

B. GROUNDS II–III: The combination of Pearce, Oran, and one of the RTP multiplexing references [Hoshi or Rosenberg] renders claims 1, 4–6, and 9–10 obvious.

As discussed in Section V.A, the combination of Pearce and Oran discloses the preambles and limitations [1.1]–[1.4], [1.7]–[1.9], [6.1]–[6.3], and [6.5]-[6.7] of claims 1 and 6. The combination further discloses the "*multiplexing*"/"*multiplexor*" limitations [1.5]/[6.4] under Patent Owner's interpretation of the plain and ordinary meaning of the term "*multiplexed stream*."

However, the combination of Pearce and Oran does not explicitly disclose these limitations under Requester's interpretation of the plain and ordinary meaning of "*multiplexed stream*" in the co-pending district court action: "a data structure containing a continuous sequence of interleaved packets of audio data from each client on the active speakers list." (*See* Appx. D (Cisco Claim Construction Brief), p. 5.) As discussed in Section V.A.2.e, the combination of Pearce and Oran teaches generation of "a continuous sequence of interleaved packets of audio data from each client on the active speakers list." However, the combination is silent on how the transmission is achieved. (Appx. C (Bress Decl.), ¶102.) Specifically, neither Pearce nor Oran explicitly discloses that the "continuous sequence of interleaved packets of audio data from each client on the active speakers list" is contained in a "data structure."

The rudimentary concept of transmitting multiplexed data in a data structure is disclosed in both Hoshi and Rosenberg as detailed below. The additional combinations presented in this ground render independent claims 1 and 6 and dependent claims 4–5 and 9–10 unpatentable as set forth below:

- GROUND II: The combination of Pearce, Oran and Hoshi renders claims 1, 5–6, and 10 unpatentable.
- GROUND III: The combination of Pearce, Oran, and Rosenberg renders claims 1, 4–6, and 9–10 unpatentable.

1. Overview of the Combinations

a. Overview of Hoshi

"Proposal of a Method of for [sic] Voice Stream Multiplexing for IP Telephony Systems" by Hoshi, et al (Appx. I ("Hoshi") was published in the IEEE Conference Proceedings for the 1999 Internet Workshop, IWS 99, held on February 18–20, 1999. (*See* Appx. R (Proceedings Front Cover); *see also* Appx. S (LOC MARC record) (indicating October 10, 1999 receipt date for proceedings in MARC entry 008).) Hoshi was also cited on the face of U.S. Patent No. 7,158,491 (Appx. T), which claims priority to U.S. Provisional Application No. 60/163,583, filed November 5, 1999, and U.S. Patent No. 8,189,592 (Appx. U), which claims priority to U.S. Provisional Application No. 60/209,551, filed June 6, 2000. Hoshi is prior art under at least pre-AIA 35 U.S.C. §§ 102(a) and (b).

Hoshi recognized that "for packet transfer over the IP network, it is necessary to add packet headers, including IP, UDP, and RTP headers and these cause an overhead resulting in inefficient bandwidth usage." (Appx. I (Hoshi), p. 182.) And moreover, "because there will be large numbers of short voice packets flowing into the IP network, the load on the Internet will increase." (Appx. I (Hoshi), p. 182.) Hoshi therefore proposed a technique for "voice stream multiplexing between IP-GWs to solve these problems." (Appx. I (Hoshi), p. 182.)

Hoshi's voice stream multiplexing "concatenate[s] RTP voice packets from the streams to the same destination IP-GWs onto a single UDP packet at a multiplexing interval period (Appx. I (Hoshi), p. 183.) Figure 3 of Hoshi (reproduced below) illustrates a "system configuration of an IP telephony system with voice stream multiplexing." (Appx. I (Hoshi), p. 184.) In this configuration, in addition to single-stream voice channels, multiplexed channels are provided between network components (e.g., IP-GWs). (Appx. I (Hoshi), p. 184.)

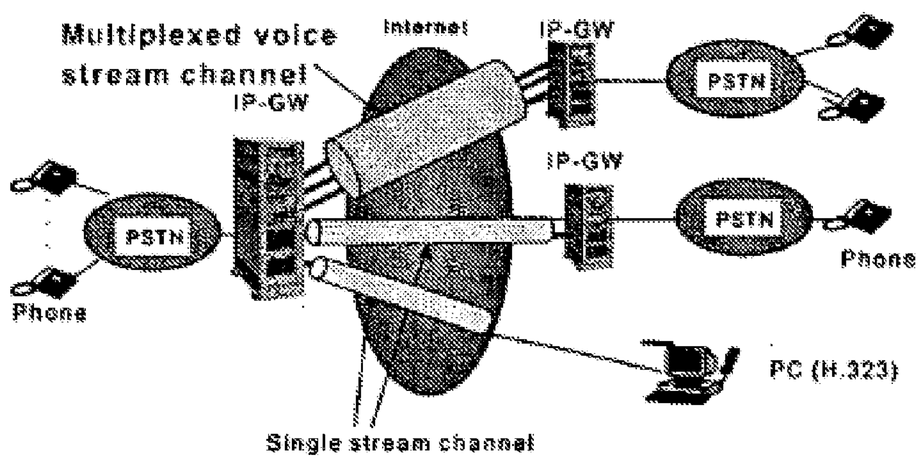
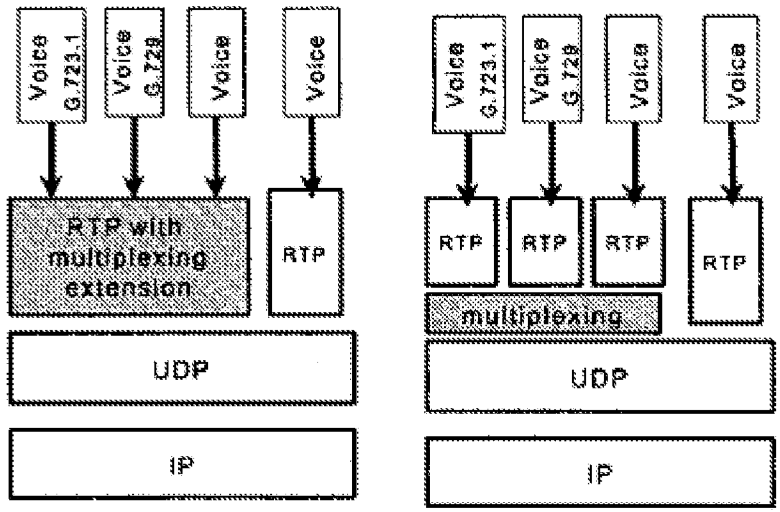


Fig. 3 IP telephony with voice stream multiplexing

Hoshi presents two techniques for its voice stream multiplexing. (Appx. I (Hoshi), p. 184, Figure 4 (below).) In the first technique, "multiplexing is done in the RTP layer so that voice frames from each voice stream are multiplexed and encapsulated into an RTP packet." (Appx. I (Hoshi), p. 184.) In the second technique, "multiplexing RTP voice packets into a UDP frame, the multiplexing layer is between the RTP and UDP layers." (Appx. I (Hoshi), p. 184.)

Fig. 4 Multiplexing layer



(a) Multiplexing in RTP (b) Multiplexing RTP packets

In Hoshi, a multiplexing interval "defines the period when RTP packets are multiplexed" to create a "multiplexed voice packet." (Appx. I (Hoshi), p. 185.) The multiplexing interval timing is a settable parameter. (Appx. I (Hoshi), p. 185.) If the timing is set to be large, "the number of RTP packets in a multiplexed packet becomes large" which "reduces both the header overhead and the number of packets, but increases the multiplexing delay." (Appx. I (Hoshi), p. 185.) If the timing is set to be small, "the number of RTP packets in a multiplexed packet becomes small, which results in smaller reductions of both header overhead and packet number." (Appx. I (Hoshi), p. 185.) After the interval timing has expired, a "multiplexed voice packet is composed by concatenating RTP-encapsulated voice packets and IP and UDP headers" as illustrated in Figure 5 below. (Appx. I (Hoshi), p. 184–185.)

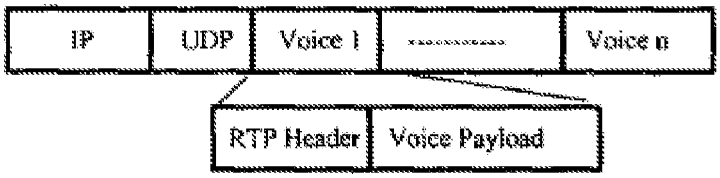


Fig. 5 Multiplexing Format

Hoshi teaches that "this method is a general RTP packet multiplexing method that is applicable not only to an IP-GW but also to other multiplexing applications." (Appx. I (Hoshi), p. 183.)

b. Overview of Rosenberg

"An RTP Payload Format for User Multiplexing" by Rosenberg, et al. ("Rosenberg"; Appx. J) is an IETF Internet Draft in the AVT Working Group dated May 6, 1998. (See Appx. J (Rosenberg).) Rosenberg was published prior to the filing date of the '858 patent. (See Appx. I (Hoshi), 188 (citing Rosenberg); Appx. BB (Internet Archive), 2 (www.ietf.org/ids.by.wg/avt.html web page listing Rosenberg and archived on February 9, 1999); Appx. CC (Internet Archive), 1 (http://www.ietf.org/internet-drafts/draft-ietf-avt-aggregation-00.txt web page for Rosenberg archived January 28, 1999).) Rosenberg was also cited in at least the following patents, having filing dates prior to the '858 patent:

- U.S. Patent No. 6,993,021 (Appx. V), filed March 8, 1999;
- U.S. Patent No. 6,704,311 (Appx. W), filed June 25, 1999;
- U.S. Patent No. 6,542,504 (Appx. X), filed May 28, 1999; and

- U.S. Patent No. 6,804,237 (Appx. Y), filed June 23, 1999.

Rosenberg is therefore prior art under at least pre-AIA 35 U.S.C. §§ 102(a) and (b).

Rosenberg describes "an RTP payload format for multiplexing data from multiple users into a single RTP packet." (Appx. J (Rosenberg), p. 1.) Rosenberg describes its technique in the context of an Internet telephony gateway (ITG) scenario, depicted in Figure 1 below. In this example, user A "wishes to speak with user C, and B wishes to speak with user D, both of which are connected to local phone network Y." (Appx. J (Rosenberg), p. 2.) To complete the call, "ITG J packetizes and transports the voice to and from A and B through the IP network, to remote gateway K" which in turn "completes the calls to C and D through PSTN Y." (Appx. J (Rosenberg), p. 2.) Rosenberg observes "that using a separate RTP session for each user connected between a pair of gateways is wasteful" because "payloads carried in each packet are generally small." (Appx. J (Rosenberg), p. 2.)

```
       ~~~~~~~~~~     -------     ~~~~~~~~~~~~     -------     ~~~~~~~~~~
A --|          |  |         |  |            |  |         |  |          |-- C
    |   PSTN   |--|   ITG   |--|   IP NET   |--|   ITG   |--|   PSTN   |
B --|    X     |  |    J    |  |            |  |    K    |  |    Y     |-- D
       ~~~~~~~~~~     -------     ~~~~~~~~~~~~     -------     ~~~~~~~~~~

 Figure 1: Internet telephony gateway architecture
```

Rosenberg therefore multiplexes the payloads from both users "into the same RTP session and packet." (Appx. J (Rosenberg), p. 2.) The format of

Rosenberg's RTP packets with multiplexed user payloads is shown in Figure 2 below. In the RTP header, the "payload type field designates the RTP packet as a multiplexed payload" and the SSRC field "is used to identify groups of users (instead of a single user) whose frames are time synchronized." (Appx. J (Rosenberg), pp. 3–4.) The multiplexing header includes user headers associated with the group of users whose media data is included in the packet. (Appx. J (Rosenberg), p.4.) The RTP packet further includes media data for the users. (Appx. J (Rosenberg), p. 5.)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                          RTP Header                           |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                          mux header                           |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      user 1 media data                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      user 2 media data                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      user 3 media data                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                                  ......
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      user N media data                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

c. **Motivation to combine**

A POSITA would have been motivated to use RTP multiplexing to transport audio data for a conference between an MCU and a transport element in the LAN

(e.g., a multicast router) in the combination of Pearce and Oran. (*See generally* Appx. C (Bress Decl.), ¶¶111–15.) Although both Hoshi and Rosenberg describe their techniques in the context of transport between Internet gateways, Hoshi suggests the use in other network elements teaching that RTP packet multiplexing "is applicable not only to an IP-GW but also to other RTP multiplexing and de-multiplexing applications." (Appx. I (Hoshi), p. 182; *see also id*., p. 183.) Rosenberg similarly "describes an RTP payload format for multiplexing data from multiple users into a single RTP packet. (Appx. J (Rosenberg), Abstract; *see also*, Conclusion ("This document has specified an RTP payload format allowing multiple user media frames to reside in an RTP packet").)

A POSITA would be further motivated to use RTP packet multiplexing in the combination of Pearce and Oran to reduce header overhead, a benefit stressed by both Hoshi and Rosenberg. (Appx. C (Bress Decl.), ¶112.) Prior to the filing date of the '858 patent, header overhead was recognized as "a key issue for voice streams in IP telephony." (Appx. I (Hoshi), p. 183.) Because of the size of the header to payload for voice packets, "only one-third of all the data is useful data (payload) and other two-thirds is overhead" making the current VoIP transfer "very inefficient." (Appx. I (Hoshi), p. 183.) Rosenberg provides an example of an ITU G.729 speech coder that "generates a rate of 8 kb/s in frames of 10 ms duration." (Appx. J (Rosenberg), p. 2) "If packed three frames per packet, the resulting RTP

payloads are 30 bytes long" and "the IP, UDP and RTP headers add up to 40 bytes, resulting in a packet efficiency of only 43%." (Appx. J (Rosenberg), p. 2) With RTP multiplexing, "the efficiency improves to 59%." (Appx. J (Rosenberg), p. 2)

A POSITA would have been further motivated to use RTP multiplexing in the combination of Pearce and Oran to reduce packetization delays. (Appx. C (Bress Decl.), ¶113; Appx. J (Rosenberg), p. 2) "Most Internet telephony applications use fairly large packetization delays, mainly for the purpose of raising the size of the payloads to increase efficiency." (Appx. J (Rosenberg), pp. 2–3.) With RTP multiplexing, "the packet payload increases" allowing "smaller packetization delays to be used as the number of multiplexed users increases." (Appx. J (Rosenberg), p. 2.)

A POSITA would have been further motivated to use RTP multiplexing to reduce interrupt processing at the network element. (Bress Decl.), ¶114; Appx. J (Rosenberg), p. 3.) Whenever a packet arrives at the gateway, "the operating system must perform a context switch into the kernel and process the packet." (Appx. J (Rosenberg), p. 3.) The "frequency of these interrupts increases linearly with the number of users." (Appx. J (Rosenberg), p. 3.) However, with RTP multiplexing, "the packet rate does not increase as more users are added, and thus the interrupt rate stays constant … improv[ing] scalability." (Appx. J (Rosenberg), p. 3.)

Finally, Pearce also provides a motivation for the combination with either Hoshi or Rosenberg, recognizing issues regarding network bandwidth: "[a]lthough integrating telecommunications into existing data networks provides many advantages, this integration does create additional network traffic which can create problems in networks with insufficient bandwidth." (Appx. G (Pearce), 1:40–44.)

2. Ground II: The combination of Pearce, Oran, and Hoshi renders claims 1, 5, 6, and 10 obvious.

a. The combination of Pearce, Oran, and Hoshi renders independent claims 1 and 6 obvious.

For the reasons discussed in Ground I, the combination of Pearce and Oran discloses limitations [1P]–[1.4] and [1.7]–[1.9] of claim 1 and limitations [6P]–[6.3], and [6.5]-[6.7] of claim 6. (*See supra* Sections V.A.2 & V.A.3.) For ease of discussion, the analysis of the limitations [1P]–[1.4], [1.6]-[1.9], [6P]–[6.3], and [6.5]-[6.7] are not repeated in this section. Hoshi discloses the "*multiplexing*"/"*multiplexor*" limitations [1.5]/[6.4] under Requestor's interpretation of the plain and ordinary meaning of "*multiplexed stream*.".

As discussed in Section II.D.1 (Claim Construction), the plain and ordinary meaning of "*multiplexed stream*" under Requestor's interpretation is "a data structure containing a continuous sequence of interleaved packets of audio data from each client on the active speakers list." The combination of Pearce, Oran, and Hoshi discloses "*(5) multiplexing said packets of audio data received from each*

client on said active speakers list into a multiplexed stream" [1.5] and "*a multiplexor capable of multiplexing said packets of audio data received from each client on said list of active speakers into a multiplexed stream*" [6.4] under this interpretation. (Appx. C (Bress Decl.), ¶¶116–23.)

As discussed in Ground I, the MCU in the combined system of Pearce and Oran identifies "*packets of audio data received from each client on said active speakers list*" to send to multicast capable devices (i.e., devices that have the capability to mix). (Appx. C (Bress Decl.), ¶117.) As discussed previously, Oran teaches that a receiver (the MCU in the combined system) "decides which speakers are currently 'active' and warrant keeping information about." (Appx. H (Oran), 3:43–44.) "Of the speakers who are identified 'active,'" the receiver "decides which speakers to mix together to produce an output audio stream." (Appx. H (Oran), 3:61–63.)

Hoshi discloses two techniques for voice stream multiplexing using RTP transport. In the first technique, "multiplexing is done in the RTP layer so that voice frames from each voice stream are multiplexed and encapsulated into an RTP packet." (Appx. I (Hoshi), p. 184.) In the second technique, "multiplexing RTP voice packets into a UDP frame, the multiplexing layer is between the RTP and UDP layers." (Appx. I (Hoshi), p. 184.) Specifically, each input stream "is sampled at a multiplexing interval by RTP packet basis, which is independent from

packetizing interval timing, and sampled RTP packets are concatenated into a multiplexed packet, which is sent to the destination IP-GW by adding UDP and IP headers." (Appx. I (Hoshi), p. 185.)

In the combined system of Pearce and Oran, utilizing Hoshi's voice stream multiplexing, the MCU multiplexes packet streams from each of the active speaker clients at a multiplexing interval. (Appx. C (Bress Decl.), ¶122.) Thus, the resulting multiplexed packet (illustrated in Hoshi Figure 5) is a data structure that contains a continuous sequence of interleaved packets of audio data from each client on the active speakers list. (Appx. C (Bress Decl.), ¶¶121–22.)

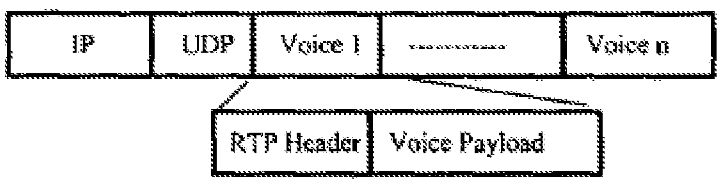


Fig. 5 Multiplexing Format

In the exemplary telecommunication session described by Pearce involving IP telephony devices 22, 23, and 25 and multicast intermediary/gateway 64a, if the multicast intermediary/gateway 64a and one IP telephony device are identified as active speakers, the multiplexed packet would contain a sequence of packets from both devices with the number of packets dependent upon the duration of the multiplexing interval. (Appx. C (Bress Decl.), ¶122.) If the timing interval is short (close to the packetization interval), the multiplexed packet includes one audio

packet from the multicast intermediary/gateway 64a and one packet from the IP telephony device. (Appx. C (Bress Decl.), ¶122.)

The combination of Pearce, Oran, and Hoshi therefore discloses or at least suggests "*(5) multiplexing said packets of audio data received from each client on said active speakers list into a multiplexed stream*" [1.5] under Requestor's interpretation. (Appx. C (Bress Decl.), ¶¶116–122.)

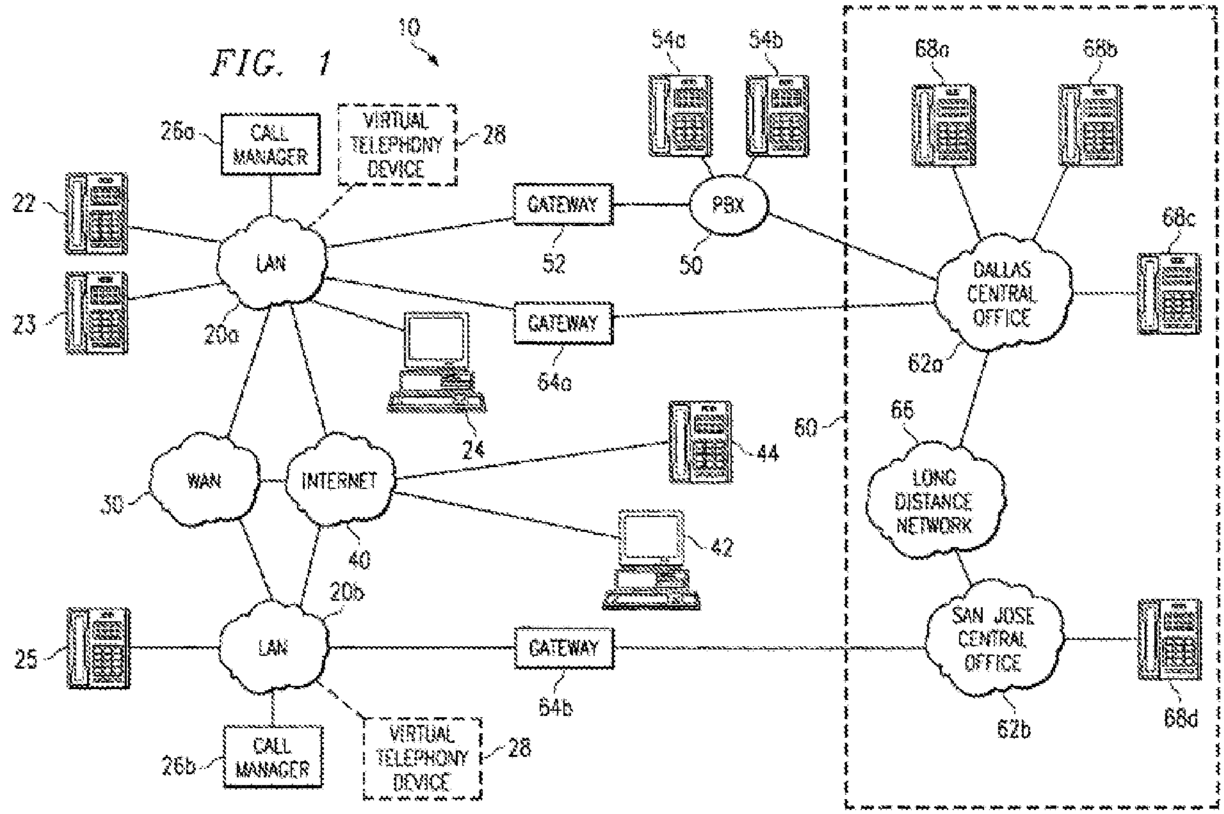
As explained in Section V.A.3.e, a POSITA would understand that, in the combination of Pearce and Oran, the MCU includes a "*multiplexor capable of*" the disclosed "*multiplexing*." A POSITA would understand that in the combination of Pearce, Oran, and Hoshi, the MCU would include a component for performing Hoshi's voice stream multiplexing. (Appx. C (Bress Decl.), ¶123.) The combination of Pearce, Oran, and Hoshi therefore also teaches or at least suggests "*a multiplexor capable of multiplexing said packets of audio data received from each client on said list of active speakers into a multiplexed stream*" [6.4] under Requestor's interpretation of "*multiplexed stream*." (Appx. C (Bress Decl.), ¶¶116–123.)

b. The combination of Pearce, Oran, and Hoshi renders dependent claims 5 and 10 obvious.

The combination of Pearce, Oran, and Hoshi discloses that "*at least one of said second subset of the plurality of clients is using a telephone and the H.323 protocol*" as recited in dependent claim 5 and that "*at least one of the plurality of*

clients, which does not have the capability to mix multiple audio streams, is using a telephone and the H.323 protocol" as recited in dependent claim 10. (Appx. C (Bress Decl.), ¶¶124–127.)

As discussed in Ground I, Pearce discloses an exemplary telecommunications session between a PSTN telephony device 68a and one or more multicast IP telephony devices 22, 23, 25, illustrated in Pearce Figure 1 below. (*See* Appx. G (Pearce), 10:55–61.) In Pearce, calls between IP telephony devices (e.g., devices 22, 23) and non-IP telephony devices 68 "are made through a gateway" (e.g., gateway 64). (Appx. G (Pearce), 5:9–12.) A gateway 64 "converts analog or digital circuit-switched data transmitted by PSTN 60 to packetized data transmitted by LAN 20, and vice-versa." (Appx. G (Pearce), 5:14–16.) Thus, the combination of Pearce and Oran discloses that "*at least one of said second subset of the plurality of clients is using a telephone*" [5]. As discussed in Section V.A.1.a, gateway 64a in Pearce is a unicast device and therefore does not have the capability to mix audio during a telecommunications session. Therefore, the combination of Pearce and Oran also discloses "*at least one of the plurality of clients, which does not have the capability to mix multiple audio streams, is using a telephone*" [10]. (Appx. C (Bress Decl.), ¶126.)



The detailed description of the ’858 patent provides only brief mentions of H.323. (*E.g.*, Appx. A (’858 patent), 4:7–11 (“The Switch 114 enables the service provider's MCU 116 to receive audio packets from both PC-based clients 102 using, for example, the SIP protocol, as well as receive H.323 protocol packets from the telephone-based clients 108 who connect via gateway 112.”), 5:29–31 (“Whether a particular party is a mixing client (e.g., a PC-based client 102 using SIP) or not (e.g., a telephone client 108 using H.323)…”).) Telephone 108 is coupled to gateway 112 through a circuit-switched network. (Appx. A (’858 patent), 3:66–41.) Solely for purposes of this reexamination, Requestor interprets

155853739.8
155853739.12

this limitation as "*the plurality of clients is using … the H.323 protocol* [via a gateway]." The combination of Pearce, Oran, and Hoshi discloses claims 5 and 10 under this interpretation.

Hoshi explains that its IP telephony system "is based on many standards such as ITU-T H.323 packet-based multimedia communication systems." (Appx. I (Hoshi), p. 182.) Hoshi notes that an "advantage of this method is that no new additional headers are required and current well-defined H.323 and RTP standards can be applied." (Appx. I (Hoshi), p. 183.) A POSITA would therefore understand that the gateway, IP telephones, multicast gateway, and MCU communicate using H.323 in the combination of Pearce, Oran, and Hoshi. (Appx. C (Bress Decl.), ¶127.) Indeed, the '858 patent acknowledges that H.323 was adopted more than 4 years before its filing date. (*See* Appx. A ('858 patent), 1:47-59.)

Accordingly, the combination of Pearce, Oran and Hoshi discloses that "*at least one of said second subset of the plurality of clients is using a telephone and the H.323 protocol* [via a gateway]" [5] and that "*at least one of the plurality of clients, which does not have the capability to mix multiple audio streams, is using a telephone and the H.323 protocol* [via a gateway]" [10] in the same manner as the '858 patent. (Appx. C (Bress Decl.), ¶¶124–27.)

3. GROUND III: The combination of Pearce, Oran, and Rosenberg renders claims 1, 4–6, and 9–10 obvious.

a. The combination of Pearce, Oran, and Rosenberg renders independent claims 1 and 6 obvious.

For the reasons discussed in Ground I, the combination of Pearce and Oran discloses limitations [1P]–[1.4] and [1.7]–[1.9] of claim 1 and limitations [6P]–[6.3] and [6.5]-[6.7] of claim 6. (*See supra* Sections V.A.2 & V.A.3.) For ease of discussion, the analysis of the limitations [1P]–[1.4], [1.6]-[1.9], [6P]–[6.3], and [6.5]-[6.7] are not repeated in this section.

The combination of Pearce, Oran, and Rosenberg discloses "*(5) multiplexing said packets of audio data received from each client on said active speakers list into a multiplexed stream*" [1.5] and "*a multiplexor capable of multiplexing said packets of audio data received from each client on said list of active speakers into a multiplexed stream*" [6.4] under Requestor's interpretation of "*multiplexed streams*." (Appx. C (Bress Decl.), ¶¶128–33.)

As discussed in Ground I, the MCU in the combined system of Pearce and Oran identifies "*packets of audio data received from each client on said active speakers list*" to send to multicast capable devices (i.e., devices that have the capability to mix). (Appx. C (Bress Decl.), ¶129.) As discussed previously, Oran teaches that a receiver (the MCU in the combined system) "decides which speakers are currently 'active' and warrant keeping information about." (Appx. H (Oran),

3:43–44.) "Of the speakers who are identified 'active,'" the receiver "decides which speakers to mix together to produce an output audio stream." (Appx. H (Oran), 3:61–63.)

Rosenberg teaches that the transmission is via a "data structure." (Appx. C (Bress Decl.), ¶¶131–32.) Rosenberg describes "an RTP payload format for multiplexing data from multiple users into a single RTP packet." (Appx. J (Rosenberg), p. 1.) The format of Rosenberg's RTP packets with multiplexed users is shown in Figure 2 below. The RTP packet includes an RTP header with a payload type field designating "the RTP packet as a multiplexed payload" and an SSRC field identifying groups of users "whose frames are time synchronized", a multiplexing header, and media data for the users in the group of users. (Appx. J (Rosenberg), pp. 3–5.)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                          RTP Header                           |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                          mux header                           |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       user 1 media data                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       user 2 media data                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       user 3 media data                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                            ......
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       user N media data                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

In the combined system of Pearce and Oran, utilizing Rosenberg's RTP multiplexing, the MCU multiplexes packet streams from each of the active speaker clients at an audio sampling rate (as discussed above). (Appx. C (Bress Decl.), ¶132.) Thus, the resulting multiplexed packet (illustrated in Rosenberg's Figure 2) is a data structure that contains a continuous sequence of interleaved packets of audio data from each client on the active speakers list. (Appx. C (Bress Decl.), ¶132.)

In the exemplary telecommunication session described by Pearce involving one of the IP telephony devices 22, 23, and 25 and multicast intermediary/gateway 64a, if the multicast intermediary/gateway 64a and one IP telephony device are identified as active speakers, the multiplexed packet would contain a sequence of

packets from both devices with the number of packets dependent upon sampling rate and frame generation rates of the clients. (Appx. C (Bress Decl.), ¶132.) If generation rates are the same, the multiplexed packet includes one audio packet from the multicast intermediary/gateway 64a and one packet from the IP telephony device. (Appx. C (Bress Decl.), ¶132.)

The combination of Pearce, Oran, and Rosenberg therefore discloses or at least suggests "*(5) multiplexing said packets of audio data received from each client on said active speakers list into a multiplexed stream*" [1.5] under Requestor's interpretation. (Appx. C (Bress Decl.), ¶¶128–32.)

As explained in Section V.A.3.e, a POSITA would understand that, in the combination of Pearce and Oran, the MCU includes a "*multiplexor capable of*" the disclosed "*multiplexing*." A POSITA would understand that in the combination of Pearce, Oran, and Rosenberg, the MCU would include a component for performing Rosenberg's RTP multiplexing. (Appx. C (Bress Decl.), ¶133.) The combination of Pearce, Oran, and Rosenberg therefore also discloses or at least suggests "*a multiplexor capable of multiplexing said packets of audio data received from each client on said list of active speakers into a multiplexed stream*" [6.4] under Requestor's interpretation. (Appx. C (Bress Decl.), ¶¶128–33.)

b. The combination of Pearce, Oran, and Rosenberg renders dependent claims 4 and 9 obvious.

The combination of Pearce, Oran, and Rosenberg discloses that "*at least one of said first subset of the plurality of clients is using PC-based equipment and the Session Initiation Protocol (SIP)*" as recited in dependent claim 4 and that "*at least one of the plurality of clients, which has the capability to mix multiple audio streams, is using PC-based equipment and the Session Initiation Protocol (SIP)*" as recited in dependent claim 9. (Appx. C (Bress Decl.), ¶¶134–36.)

As discussed in Ground I, Pearce discloses an exemplary telecommunications session between a PSTN telephony device 68a and one or more multicast IP telephony devices 22, 23, 25, illustrated in Pearce Figure 1 below. (*See* Appx. G (Pearce), 10:55–61.) The '858 patent equates PC-based clients to clients "which connect to a wide area network (e.g., the public internet)." (*See* Appx. A ('858 patent), 3:58-63.) The '858 patent also equates PC-based client to clients that have the capability to mix. (*See, e.g.*, Appx. A ('858 patent), 2:18-22 ("… whereby mixing (e.g., PC-based clients) and non-mixing (e.g., phone) clients can simultaneously participate in a single audio conference application".) A POSITA would understand that Pearce's multicast IP telephony devices that have the capability to perform mixing locally are each "*PC-based equipment*." (Appx. C (Bress Decl.), ¶135 (citing Appx. G (Pearce), 3:55–58).) Thus, the combination of Pearce and Oran discloses "*at least one of said first subset of the plurality of clients*

is using PC-based equipment" [4] and "*at least one of the plurality of clients, which has the capability to mix multiple audio streams, is using PC-based equipment*" [9]. (Appx. C (Bress Decl.), ¶¶134–35.)

Rosenberg notes that its gateway devices "can signal calls using SIP [1], H.323 or proprietary signalling [sic] protocols." (Appx. J (Rosenberg), p. 2.) Rosenberg also teaches that "signaling protocols such as SIP [6] together with a session description protocol such as SDP or H.323 could be used" to bind payload types to particular codec types and length values. (Appx. J (Rosenberg), p. 5.) Based on Rosenberg's disclosure, a POSITA would understand that SIP would also be used by IP telephony devices for signaling. (Appx. C (Bress Decl.), ¶136; *see also* Appx J (Rosenberg), p. 8 ("The multiplexing protocol can make use of whatever encryption and authentication schemes are present in RTP, SIP, H.323 or other relevant protocols").) Indeed, the '858 patent acknowledges that SIP was a well-known "signaling protocol for Internet conferencing and telephony" prior to its filing date. (*See* Appx. A ('858 patent), 1:61-66.)

Accordingly, the combination of Pearce, Oran, and Rosenberg discloses that "*at least one of said first subset of the plurality of clients is using PC-based equipment and the Session Initiation Protocol (SIP)*" [4] and that "*at least one of the plurality of clients, which has the capability to mix multiple audio streams, is*

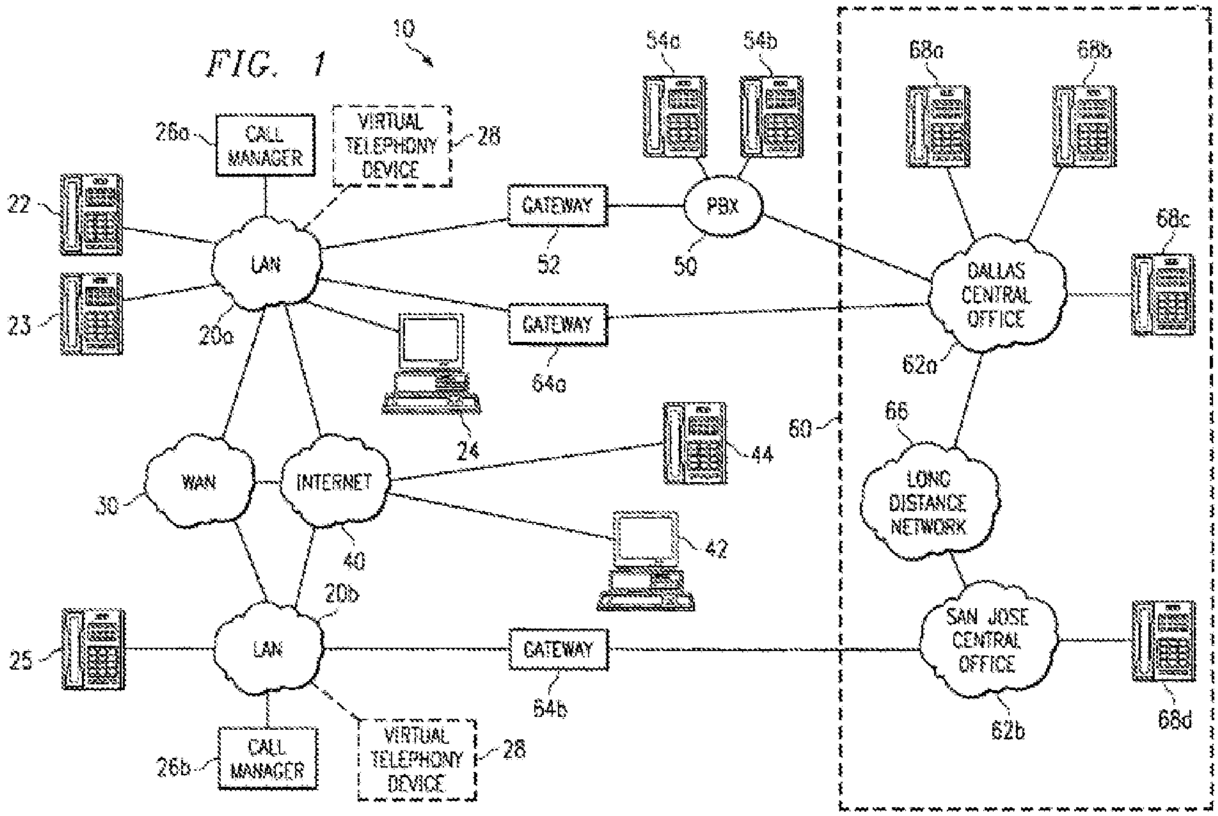
using PC-based equipment and the Session Initiation Protocol (SIP)" [9]. (Appx. C (Bress Decl.), ¶¶134–36.)

c. The combination of Pearce, Oran, and Rosenberg renders dependent claims 5 and 10 obvious.

The combination of Pearce, Oran and Rosenberg discloses that "*at least one of said second subset of the plurality of clients is using a telephone and the H.323 protocol*" as recited in dependent claim 5 and that "*at least one of the plurality of clients, which does not have the capability to mix multiple audio streams, is using a telephone and the H.323 protocol*" as recited in dependent claim 10. (Appx. C (Bress Decl.), ¶¶137–40.)

As discussed in Ground I, Pearce discloses an exemplary telecommunications session between a PSTN telephony device 68a and one or more multicast IP telephony devices 22, 23, 25, illustrated in Pearce Figure 1 below. (*See* Appx. G (Pearce), 10:55–61.) In Pearce, calls between IP telephony devices (e.g., devices 22, 23) and non-IP telephony devices 68 "are made through a gateway" (e.g., gateway 64). (Appx. G (Pearce), 5:9–12.) A gateway 64 "converts analog or digital circuit-switched data transmitted by PSTN 60 to packetized data transmitted by LAN 20, and vice-versa." (Appx. G (Pearce), 5:14–16.) Thus, the combination of Pearce and Oran discloses that "*at least one of said second subset of the plurality of clients is using a telephone*" [5]. As discussed in Section V.A.1.a, gateway 64a in Pearce is a unicast device and therefore does not have the

capability to mix audio during a telecommunications session. Therefore, the combination of Pearce and Oran also discloses "*at least one of the plurality of clients, which does not have the capability to mix multiple audio streams, is using a telephone*" [10]. (Appx. C (Bress Decl.), ¶139.)



Rosenberg notes that its gateway devices "can signal calls using SIP [1], H.323 or proprietary signalling [sic] protocols." (Appx. J (Rosenberg), p. 2.) Rosenberg also teaches that "signaling protocols such as SIP [6] together with a session description protocol such as SDP or H.323 could be used" to bind payload types to particular codec types and length values. (Appx. J (Rosenberg), p. 5.) A

POSITA would therefore understand that the gateway, IP telephones, multicast gateway, and MCU communicate using H.323 in the combination of Pearce, Oran, and Hoshi. (Appx. C (Bress Decl.), ¶140.) Indeed, the '858 patent acknowledges that H.323 was adopted more than 4 years before its filing date. (*See* Appx. A ('858 patent), 1:47-59.) Rosenberg therefore teaches that "*at least one of said second subset of the plurality of clients is using … the H.323 protocol* [via a gateway]" [5] and "*at least one of the plurality of clients, which does not have the capability to mix multiple audio streams, is using … the H.323 protocol* [via a gateway]" [10] in the same manner as discussed in the '858 patent. (Appx. C (Bress Decl.), ¶140.)

Accordingly, the combination of Pearce, Oran, and Rosenberg discloses that "*at least one of said second subset of the plurality of clients is using a telephone and the H.323 protocol* [via a gateway]" [5] and that "*at least one of the plurality of clients, which does not have the capability to mix multiple audio streams, is using a telephone and the H.323 protocol* [via a gateway]" [10]. (Appx. C (Bress Decl.), ¶137–40.)

C. GROUNDS IV–VI: The combination of Pearce, Oran, [Hoshi or Rosenberg] and Robert renders dependent claims 2–3 and 7–8 obvious.

Claims 2–3 (which depend from claim 1) and 7–8 (which depend from claim 6) recite, generally, the removal of one client's packets from either the multiplexed

stream or the combined packet before sending to that client. As discussed above in Ground I, the combination of Pearce and Oran renders claims 1 and 6 obvious under Patent Owner's interpretation of "*multiplexed stream*." As discussed above in Grounds II–III, the combination of Pearce and Oran with either of the RTP multiplexing references (Hoshi or Rosenberg) renders claims 1 and 6 obvious under Requestor's (and Patent Owner's) interpretation of "*multiplexed stream*." However, none of those references explicitly discloses the subject matter of dependent claims 2–3 and 7–8. Robert provides these teachings.

The following grounds therefore render claims 2–3 and 7–8 unpatentable:

- GROUND IV: The combination of Pearce, Oran and Robert
- GROUND V: The combination of Pearce, Oran, Hoshi, and Robert
- GROUND VI: The combination of Pearce, Oran, Rosenberg, and Robert

1. Overview of the Combination

a. Overview of Robert

The application which issued as United States Patent No. 6,327,276 to Robert, et al. ("Robert"; Appx. K) was filed on December 22, 1998. Robert is therefore prior art under at least pre-AIA 35 U.S.C. § 102(e).

Robert relates to a method and system "for audio or other types of conferencing over a LAN/WAN using multicasting where the conferencing function is distributed between the server and the clients." (Appx. K (Robert), 1:8–

12, *see also,* 3:40–44.) Robert explains that the "[c]urrent methods of facilitating conferencing of users over a LAN/WAN are either client-based or server-based." (Appx. K (Robert), 1:35–36.) In a client-based system, illustrated in Figure 1 below, "each of the clients in the conference call transmits their signals over the LAN/WAN to every other client involved in the conference call." (Appx. K (Robert), 3:66–4:2.) Each client "in the conference call receives individual signals from all of the other clients involved in the call." (Appx. K (Robert), 4:2–4.) Combining the individually received signals into an audio stream for the user is "performed locally, at each of the receiving clients." (Appx. K (Robert), 4:4–6.) Robert notes that a drawback to this client-based system is that "[m]uch network bandwidth is used in the transmission of signals from each client to every other client in a conference call." (Appx. K (Robert), 4:30–34.)

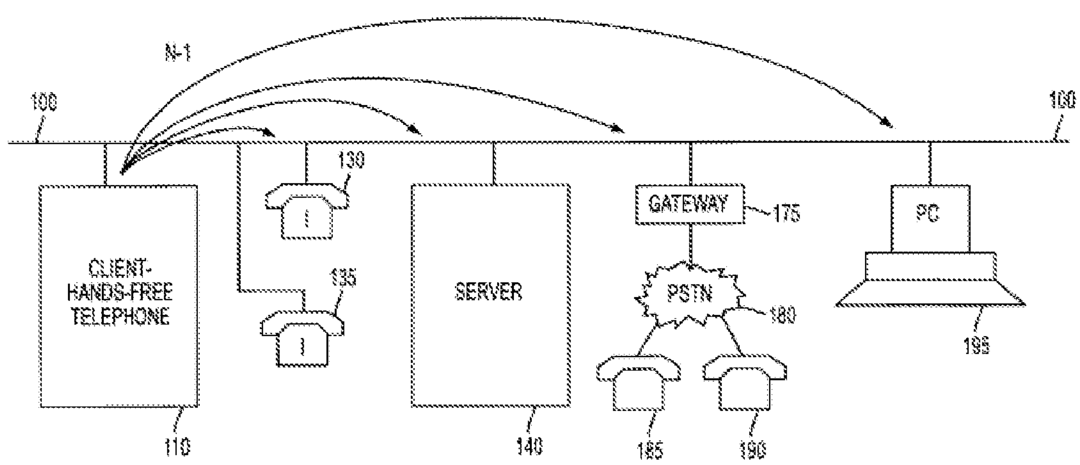


FIG. 1
(PRIOR ART)

In a server-based system, illustrated in Robert's Figure 2 below, "the conferencing function is performed entirely by the server 240." (Appx. K (Robert), 4:48–50.) Each client involved in a conference call transmits its individual signals to the server which "then combines them with additional loss applied to the signals of inactive (not talking) participants so as to minimize additive noise in the output signals." (Appx. K (Robert), 4:50–54.) The server then "transmits a client specific signal to each of the clients." (Appx. K (Robert), 4:54–55.)

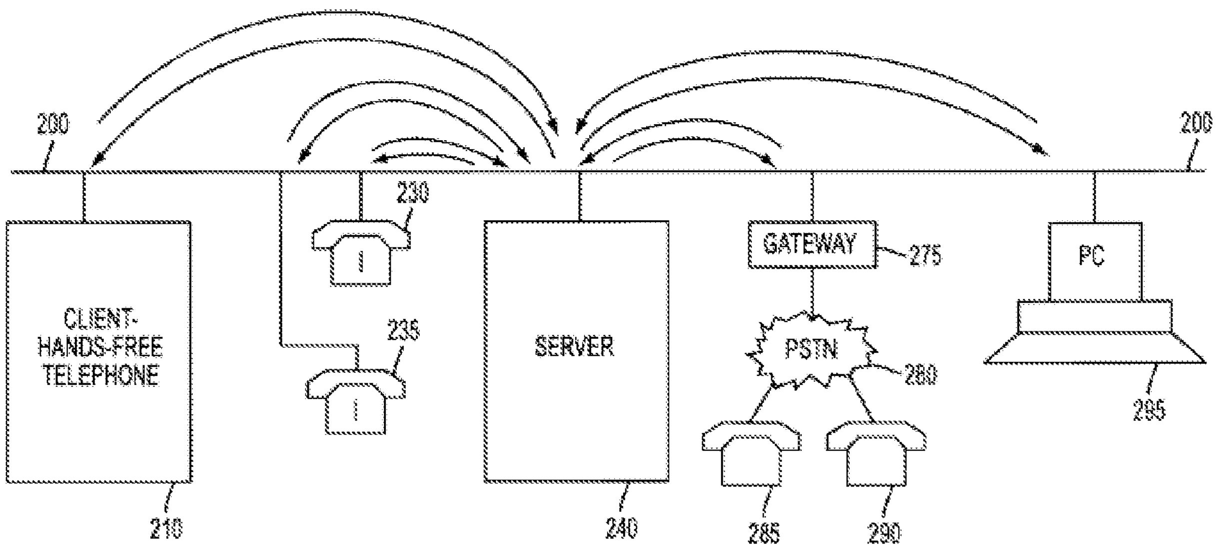


FIG. 2
(PRIOR ART)

In the prior-art server-based system of Figure 2, the combined signal sent to a particular client contains "the combined signals of the other clients **but does not contain that particular client's own signal**." (Appx. K (Robert), 4:63–65.) Robert notes that "[u]sually in a conference call, only one or two of the parties are speaking at any given time" and "the remaining parties are relatively silent."

(Appx. K (Robert), 5:2–4.) Therefore, "in the server-based conferencing system, at any given time only two of the signals received by the server are 'active'." (Appx. K (Robert), 5:4–6.) Because of this, "those clients in the voice conference call who are not talking all receive separate but identical combined signals from the server." (Appx. K (Robert), 5:6–9.) Robert notes that "[o]ne drawback of the server-based method this method [sic] is that it results in unnecessary network traffic, leading to possible congestion and delay." (Appx. K (Robert), 5:11–13.)

To address the drawbacks of the client-based and server-based systems, Robert describes "a hybrid client/server system for distributed conferencing" illustrated in Figure 3 below. In this hybrid system, "[e]ach of the clients involved in a conference call transmits their individual signals to the server 340." (Appx. K (Robert), 5:51–53.) These "individual transmit signals are then scaled and mixed at the server 340 to create a single multicast signal, which is then transmitted to each of the clients." (Appx. K (Robert), 5:53–56.) Each client in the hybrid client/server system "receives the same signal." (Appx. K (Robert), 5:56–57.) "A particular client receiving the multicast signal then removes its own component from the signal and outputs the remaining signal which contains the conferenced signals of the other users in the conference call but substantially no 'echo' of that particular client's output signal." (Appx. K (Robert), 5:57–61.)

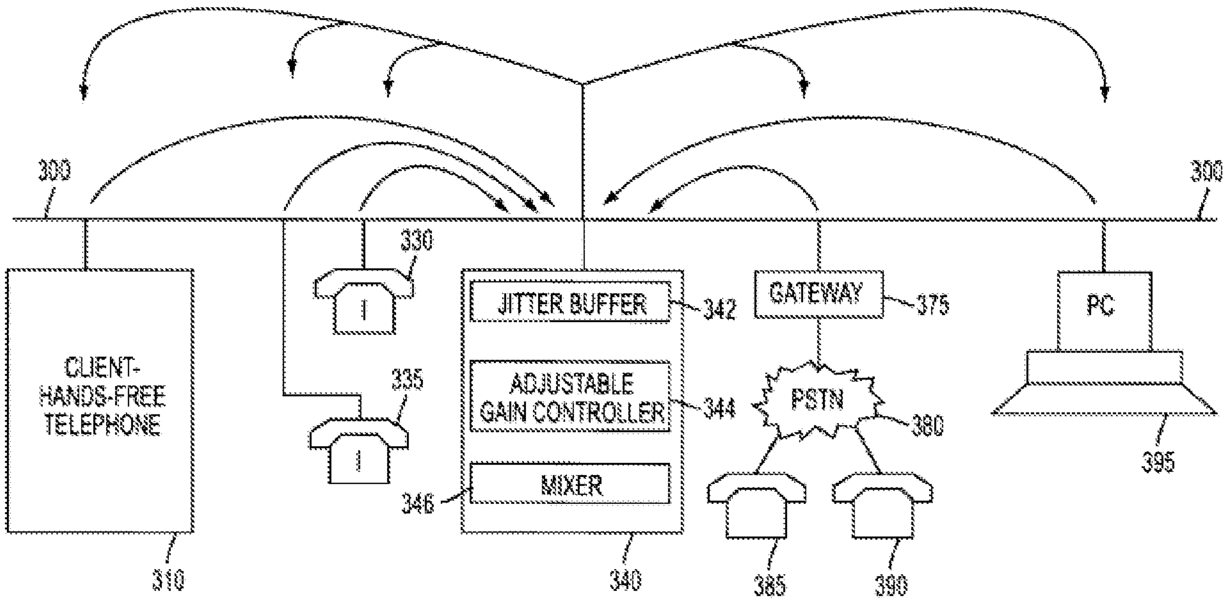


FIG. 3

b. Motivation to Combine

A POSITA would have been motivated to combine Robert's teaching of echo control into the conferencing system of Pearce and Oran. (Appx. C (Bress Decl.), ¶146.) A POSITA would recognize that an active speaker that receives either a multiplexed stream with their own audio data to mix locally, or a mixed packet that included their own audio data, to play would experience a form of "echo" because they will hear their own voice after a period of delay from when they spoke. (Appx. C (Bress Decl.), ¶146.) A POSITA would have therefore been motivated to remove a speaker's own audio from either the "*multiplexed stream*" or "*combined packet*" to eliminate the echo artifact and provide a better user experience. (Appx. C (Bress Decl.), ¶146.) Indeed, such forms of echo cancellation

155853739.8
155853739.12

were well known prior to the ’858 patent. (Appx. C (Bress Decl.), ¶146 (citing Appx. K (Robert), 2:41–45 (“When a particular client receives the transmitted multicast signal, the signal component corresponding to the signal transmitted from that client is removed so that the remaining signal contains substantially no ‘echo’ of the client’s own signal.”); Appx. O (O’Malley), 2:4–9 (“The centralized audio mixer provides the summed conference signal to the audio processor(s) for post processing and routing to the conference participants. The post processing includes removing the audio associated with a speaker from the conference signal to be sent to the speaker.”).)

2. The combination of Pearce, Oran, [Hoshi or Rosenberg] and Robert renders dependent claims 2 and 7 obvious.

The combination of Pearce, Oran, and Robert, or Pearce, Oran, [Hoshi or Rosenberg] and Robert, discloses or suggests “*before sending said multiplexed stream to one of said first subset of the plurality of clients, removing from said multiplexed stream said packets of audio data received from said one of said first subset of the plurality of clients when said one of said first subset of the plurality of clients is on said active speakers list*” as recited in dependent claim 2 (Appx. C (Bress Decl.), ¶147–49) and “*means for removing, before said packet sender sends said multiplexed stream to one of the plurality of clients which have the capability to mix multiple audio streams, from said multiplexed stream said packets of audio*

data received from said one of the plurality of clients, when said one of the plurality of clients is on said list of active speakers" as recited in claim 7.

Robert discloses that a client receiving a multicast, composite audio signal "**removes its own component** from the signal and outputs the remaining signal which contains the conferenced signals of the other users in the conference call but substantially no 'echo' of that particular client's output signal." (Appx. K (Robert), 5:57–61.) In the combined system, the multicast signal sent by the MCU is a "*multiplexed stream*," as discussed in Ground I. A POSITA would understand that in the "*multiplexed stream*" a client's "own component" is the audio packet or packets that the client transmitted. (Appx. C (Bress Decl.), ¶148.) Therefore, as discussed in the motivation to combine section, based on these teachings of Robert of removing a client's own component from a received mixed audio signal, a POSITA would have been motivated to remove a client's own "components" (i.e., audio packets) from the "*multiplexed signal*." (Appx. C (Bress Decl.), ¶148.) Thus, the combination of Pearce, Oran, and Robert, or Pearce, Oran, [Hoshi or Rosenberg] and Robert, discloses "*removing from said multiplexed stream said packets of audio data received from said one of said first subset of the plurality of clients when said one of said first subset of the plurality of clients is on said active speakers list*" [2] (Appx. C (Bress Decl.), ¶¶147–48.) and "*means for removing … from said multiplexed stream said packets of audio data received from*

said one of the plurality of clients, when said one of the plurality of clients is on said list of active speakers" [7].

A POSITA would have been motivated to perform the "*removing*" step at multiple places in the network. In the combination of Pearce, Oran, and Robert and in the combinations of Pearce, Oran, (Hoshi or Rosenberg), and Robert, a POSITA would have been motivated to perform the "*removing*" step at a client device before "*sending*" the audio packets in the received "*multiplexed stream*" to an application for mixing. (Appx. C (Bress Decl.), ¶149.) This option takes full advantage of the benefits of multicasting and addresses the risk of echo created by mixing in a client's own audio stream. (Appx. C (Bress Decl.), ¶149.) And in the combination of Pearce, Oran and Robert, a POSITA would have been motivated to perform the "*removing*" step at the last multicast router before the "*multiplexed stream*" (under Patent Owner's interpretation) is sent to the end user client device. (Appx. C (Bress Decl.), ¶149.) Indeed, this option was a defined part of the multicasting standard prior to the '858 patent. (Appx. C (Bress Decl.), ¶149, *citing* Appx. AA (IETF RFC 2462). In both options, a client's own packets of audio data are removed before sending the "*multiplexed stream*" to the client or client conferencing application.

Accordingly, the combination of Pearce, Oran, and Robert, or Pearce, Oran, [Hoshi or Rosenberg] and Robert, discloses or suggests "*before sending said*

multiplexed stream to one of said first subset of the plurality of clients, removing from said multiplexed stream said packets of audio data received from said one of said first subset of the plurality of clients when said one of said first subset of the plurality of clients is on said active speakers list" [2] (Appx. C (Bress Decl.), ¶¶147–49) and "*means for removing, before said packet sender sends said multiplexed stream to one of the plurality of clients which have the capability to mix multiple audio streams, from said multiplexed stream said packets of audio data received from said one of the plurality of clients, when said one of the plurality of clients is on said list of active speakers*" [7]

3. The combination of Pearce, Oran, [Hoshi or Rosenberg] and Robert renders dependent claims 3 and 8 obvious.

The combination of Pearce, Oran, and Robert, or Pearce, Oran, [Hoshi or Rosenberg] and Robert, discloses "*before sending said combined packet to one of said second subset of the plurality of clients, removing from said combined packet said packets of audio data received from said one of said second subset of the plurality of clients when said one of said second subset of the plurality of clients is on said active speakers list*" as recited in claim 3 (Appx. C (Bress Decl.), ¶¶150–52.) and "*means for removing, before said packet sender sends said combined packet to one of the plurality of clients which do not have the capability to mix multiple audio streams, from said combined packet said packets of audio data*

received from said one of the plurality of clients, when said one of the plurality of clients is on said list of active speakers" as recited in claim 8.

Robert discloses that a client receiving a multicast signal having mixed audio "removes its own component from the signal and outputs the remaining signal which contains the conferenced signals of the other users in the conference call but substantially no 'echo' of that particular client's output signal." (Appx. K (Robert), 5:57–61.) In the combined system, the multicast intermediary transmits a "*combined packet*" containing the mixed audio packets "*of audio data received from each client on said active speakers* list" as discussed in Ground I. A POSITA would understand that in the "*combined packet*" a client's "own component" is its contribution to the mixed audio. (Appx. C (Bress Decl.), ¶151.) Therefore, as discussed in the motivation to combine section, based on these teachings of Robert, a POSITA would have been motivated to remove a client's own "components" (i.e., the client's contribution to the mixed audio) from the "*combined signal*." (Appx. C (Bress Decl.), ¶151.)

Thus, the combination Pearce, Oran, and Robert, or Pearce, Oran, [Hoshi or Rosenberg] and Robert, discloses "*removing from said combined packet said packets of audio data received from said one of said second subset of the plurality of clients when said one of said second subset of the plurality of clients is on said active speakers list*" [3] (Appx. C (Bress Decl.), ¶¶150–51.) and "*means for*

removing … from said combined packet said packets of audio data received from said one of the plurality of clients, when said one of the plurality of clients is on said list of active speakers" [8].

A POSITA would have been motivated to perform the "*removing*" step at the multicast intermediary before the modified "*combined packet*" is sent to the gateway 64a for conversion and transmission to the PSTN telephony device 68a. (Appx. C (Bress Decl.), ¶152.) This option takes full advantage of the benefits of the use of a multicast intermediary and addresses the risk of echo created by mixing in a client's own audio stream. (Appx. C (Bress Decl.), ¶152.)

Accordingly, the combination Pearce, Oran, and Robert, or Pearce, Oran, [Hoshi or Rosenberg] and Robert, discloses "*before sending said combined packet to one of said second subset of the plurality of clients, removing from said combined packet said packets of audio data received from said one of said second subset of the plurality of clients when said one of said second subset of the plurality of clients is on said active speakers list*" [3] (Appx. C (Bress Decl.), ¶¶150–52.) and "*means for removing, before said packet sender sends said combined packet to one of the plurality of clients which do not have the capability to mix multiple audio streams, from said combined packet said packets of audio data received from said one of the plurality of clients, when said one of the plurality of clients is on said list of active speakers*" [8].

D. GROUNDS VII–IX: The combination of Pearce, Oran, [Hoshi or Rosenberg] and Salama renders dependent claims 4–5 and 9–10 obvious.

Claims 4–5 (which depend from claim 1) and 9–10 (which depend from claim 6) recite, generally, at least one client using PC-based equipment and the Session Initiation Protocol (SIP) (claims 4 and 9) and at least one client using a telephone and the H.323 protocol (claims 5 and 10). However, the combination of Pearce and Oran, alone, does not explicitly disclose the subject matter of dependent claims 4–5 and 9–10. Salama provides these teachings.

The following grounds therefore render claims 4–5 and 9–10 unpatentable:

- GROUND VII: The combination of Pearce, Oran, and Salama
- GROUND VIII: The combination of Pearce, Oran, Hoshi, and Salama
- GROUND IX: The combination of Pearce, Oran, Rosenberg, and Salama

1. Overview of the Combination

a. Overview of Salama

The application which issued as United States Patent No. 6,584,093 (“Salama”; Appx. L) was filed January 5, 1999 and issued June 24, 2003. Salama claims priority to United States Provisional Application No. 9786698, which was filed August 25, 1998. Salama is prior art under at least pre-AIA 35 U.S.C. § 102(e).

Salama is directed to the "inter-domain routing of calls in a network." (Appx. L (Salama), Abstract.) Salama explains that "Internet Telephony allows telephone calls to be carried over an Internet protocol (IP) network either end-to-end between two telephones or computers, or as one or more 'hops' in an end-to-end telephone call." (Appx. L (Salama), 1:14–17.) Salama recognized that in order "to reduce the cost of voice calls," "a voice call may be to be routed over multiple hops, with some of these hops being in the data network while others are in the voice network." (Appx. L (Salama), 1:18–23.) Salama thus proposed a "mechanism for selecting the best path towards the destination address," which may be "a PSTN phone, an IP phone, or any other voice terminal." (Appx. L (Salama), 9:1–4.) Salama's solution leverages two standard signaling protocols that were evolving at the time of the invention: H.323 and SIP. (Appx. L (Salama), 1:14–17.) As Salama explains, "signaling protocols, when combined with a method of routing the signaling messages and maintaining call state[,] allow the actual media (i.e. voice) to flow in packets between the endpoints." (Appx. L (Salama), 1:25–28.)

For example, Figure 10 of Salama (reproduced below) is identified in the specification as prior art and illustrates "an example of an Internet Telephony call from an SIP terminal through the IP network composed of multiple ISPs to another SIP terminal wherein a series of SIP proxy servers control routing of the call."

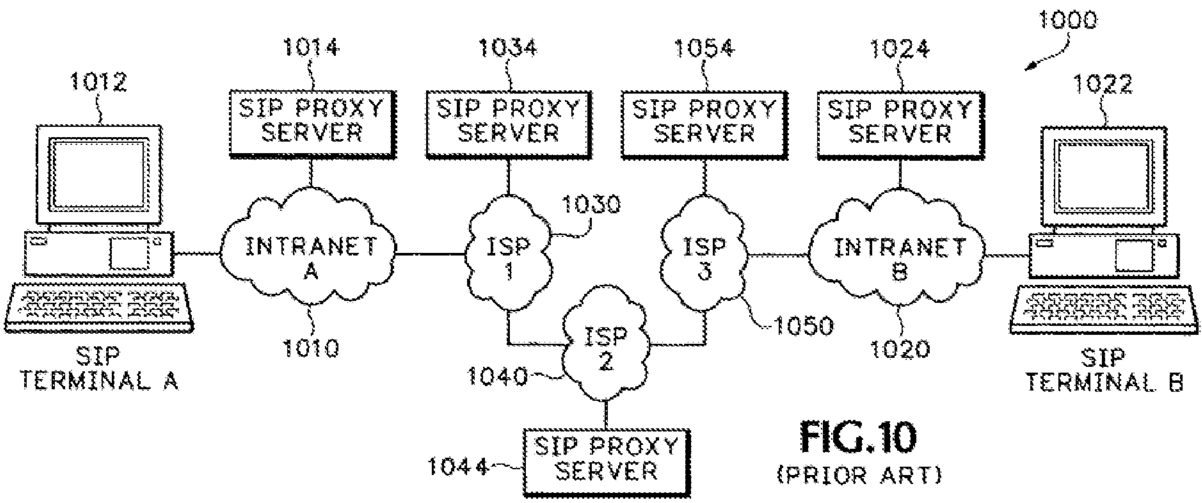


FIG.10
(PRIOR ART)

Additionally, Figure 13 of Salama (reproduced below) is identified in the specification as prior art and illustrates "an example of an Internet Telephony call from a PSTN terminal through an SIP/PSTN gateway to the IP network to an SIP/H.323 gateway onto an intranet and to a H.323 terminal." (Appx. L (Salama), 8:30–33.) In this example, the SIP/H.323 gateway 1326 allows the H.323 protocol to be used with a PSTN terminal. (Appx. L (Salama), 6:37–43 ("SIP/H.323 gateway 1326 [] can convert SIP protocol calls received from gateway 1116 over IP network 1130 into H.323 protocol calls which are routed over intranet 1320 to

H.323 terminal 1322. SIP/H.323 gateway also converts H.323 calls received over intranet 1320 into SIP for transmission over internet 1130 to gateway 1116.").) Here, Salama identified an outstanding need "to keep track of which signaling protocol should be used on a particular segment of the IP networks." (Appx. L (Salama), 6:43–46.)

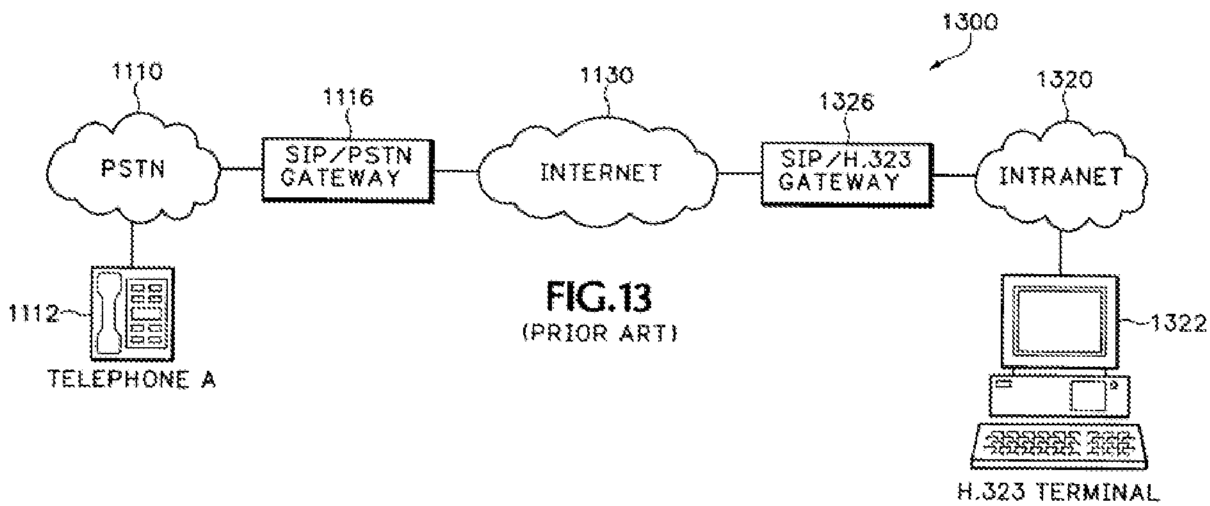


FIG.13
(PRIOR ART)

b. Motivation to Combine

A POSITA would have been motivated to combine Salama's teachings of the use of the SIP and H.323 protocols with the combined audio conferencing system of Pearce and Oran. (*See generally* Appx. C (Bress Decl.), ¶¶157–58.) First, Pearce motivates the combination stating that the system uses "TCP signaling 102 (**or any other appropriate type of signaling**)" with the call manager for call setup and control. (Appx. G (Pearce), 10:55-61.) SIP and H.323 were two widely known and implemented types of signaling for packet-based

155853739.8
155853739.12

communication. Salama explains that, prior to its filing date of January 1999, SIP was a known "signaling protocol for establishing connections between endpoints participating in a conference call." (Appx. L (Salama, 5:45–48.) Similarly, H.323 was already "a standard architecture for multimedia conferencing (voice, video, and, data) in packet-based networks that was designed by the ITU-T." (Appx. L (Salama, 2:9–11.) And Salama notes that before its filing date, "H.323 ha[d] been successfully applied as a suite of signaling protocols for Internet Telephony." (Appx. L (Salama, 2:11–13.)

A POSITA would have been motivated to use Salama's method of call routing which utilizes SIP and H.323 to reduce the costs of calls. (Appx. C (Bress Decl.), ¶157.) Salama's technique (described above) selects "the best path towards the destination address," which may be "a PSTN phone, an IP phone, or any other voice terminal." (Appx. L (Salama), 9:1–4.) By selecting the optimal path, a POSITA would recognize that network traffic is reduced, saving bandwidth and reducing the cost of calls. Just as a POSITA would have been motivated to develop a system that "reduces network traffic and saves network bandwidth" (Appx. G (Pearce, 2:17–24) and overcomes system limitations associated with processing resources and memory (Appx. H (Oran, 3:26–30 ("If processing and memory were not a concern, state information could be kept at each receiver 24A–24D for all potential speakers 26A-26D, and for all those speakers who might be talking at any

given time."), 1:25–26 (describing the "limited processing resources of individual receivers")), a POSITA have been also been motivated to enhance the combined system by reducing the cost but maintaining the quality of such audio conferencing. (Appx. C (Bress Decl.), ¶157.) Indeed, a "major objective in creating an internet telephony system is to reduce the cost of voice calls while maintaining the same quality level currently provided in voice networks." (Appx. L (Salama), 1:17–20.)

2. The combination of Pearce, Oran, [Hoshi or Rosenberg] and Salama renders dependent claims 4 and 9 obvious.

The combination of Pearce, Oran, and Salama, or Pearce, Oran, [Hoshi or Rosenberg], and Robert, discloses that "*at least one of said first subset of the plurality of clients is using PC-based equipment and the Session Initiation Protocol (SIP)*" as recited in dependent claim 4 and that "*at least one of the plurality of clients, which has the capability to mix multiple audio streams, is using PC-based equipment and the Session Initiation Protocol (SIP)*" as recited in dependent claim 9. (Appx. C (Bress Decl.), ¶¶159–61.)

As discussed in Ground I, Pearce discloses an exemplary telecommunications session between a PSTN telephony device 68a and one or more multicast IP telephony devices 22, 23, 25, illustrated in Pearce Figure 1 below. (*See* Appx. G (Pearce), 10:55–61.) The '858 patent prior art equates PC-based clients to clients "which connect to a wide area network (e.g., the public

internet).” (*See* Appx. A (’858 patent), 3:58-63.) The ’858 patent also equates PC-based client to clients that have the capability to mix. (*See, e.g.,* Appx. A (’858 patent), 2:18-22 (“… whereby mixing (e.g., PC-based clients) and non-mixing (e.g., phone) clients can simultaneously participate in a single audio conference application”.) A POSITA would understand that Pearce’s multicast IP telephony devices that have the capability to perform mixing locally are each “*PC-based equipment*.” (Appx. C (Bress Decl.), ¶160 (citing Appx. G (Pearce), 3:55–58).) Therefore, the combination or Pearce and Oran also discloses “*at least one of the plurality of clients, which has the capability to mix multiple audio streams, is using PC-based equipment*” [9]. (Appx. C (Bress Decl.), ¶¶159–60.)

Salama explains that, at the time of the invention, the “Session Initiation Protocol (SIP) is an Internet Conferencing Protocol being developed in the IETF. SIP is a signaling protocol for establishing connections between endpoints participating in a conference call.” (Appx. L (Salama), 5:45–48.) As depicted in Figure 10 of Salama, a POSITA would have understood that the SIP protocol could

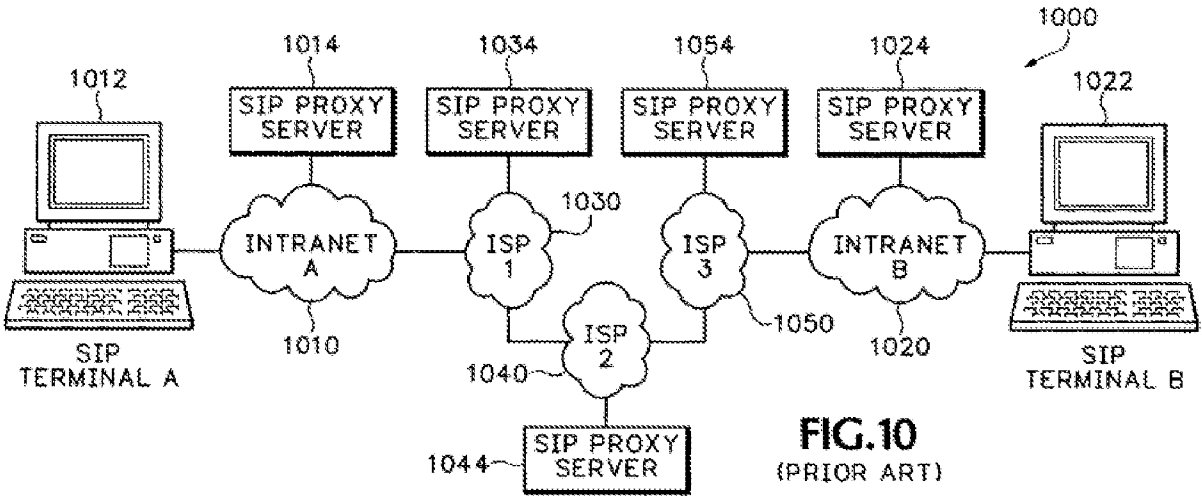


FIG.10
(PRIOR ART)

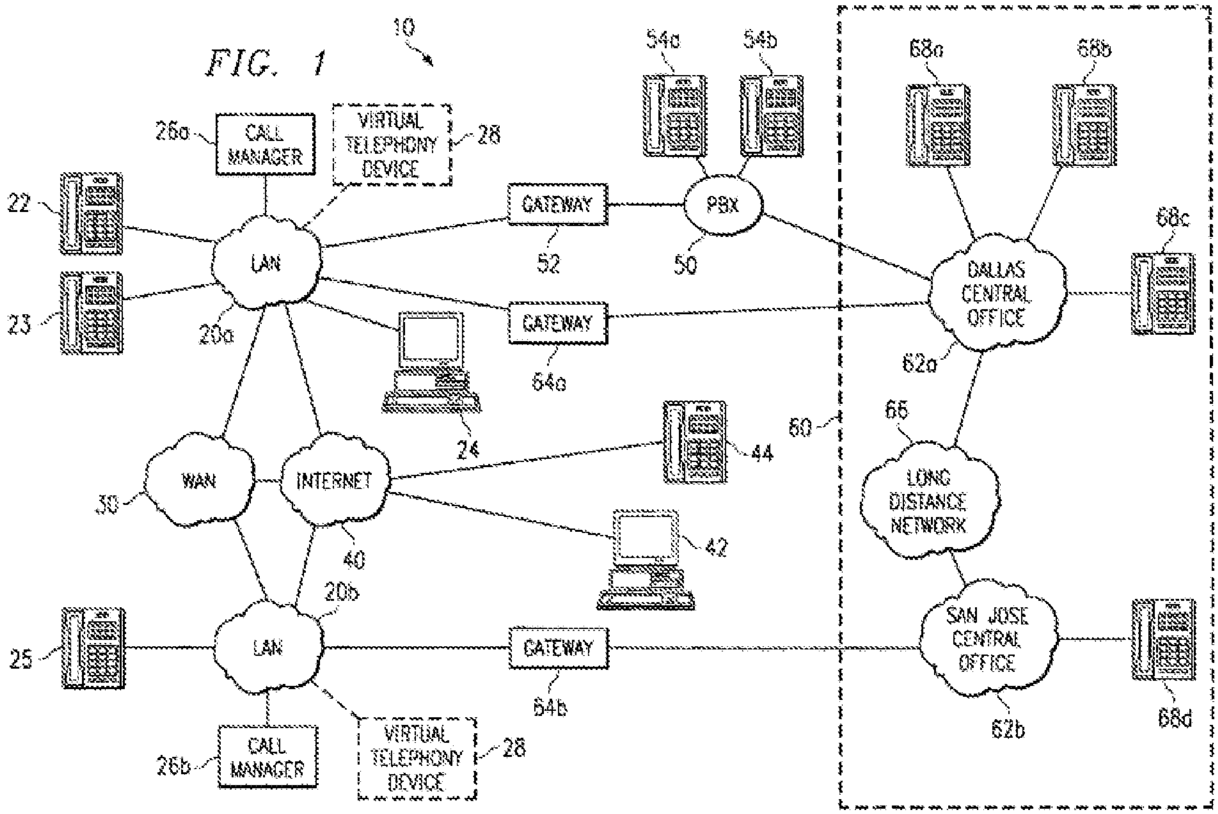
Accordingly, the combination of Pearce, Oran, and Salama, or Pearce, Oran, [Hoshi or Rosenberg], and Salama, discloses that "*at least one of said first subset of the plurality of clients is using PC-based equipment and the Session Initiation Protocol (SIP)*" [4] and that "*at least one of the plurality of clients, which has the capability to mix multiple audio streams, is using PC-based equipment and the Session Initiation Protocol (SIP)*" [9]. (Appx. C (Bress Decl.), ¶¶159–61.)

3. The combination of Pearce, Oran, [Hoshi or Rosenberg] and Salama renders dependent claims 5 and 10 obvious.

The combination of Pearce, Oran, and Salama, or Pearce, Oran, [Hoshi or Rosenberg], and Salama, discloses that "*at least one of said second subset of the plurality of clients is using a telephone and the H.323 protocol*" as recited in dependent claim 5 and that "*at least one of the plurality of clients, which does not*

have the capability to mix multiple audio streams, is using a telephone and the H.323 protocol" as recited in dependent claim 10. (Appx. C (Bress Decl.), ¶¶162–65.)

As discussed in Ground I, Pearce discloses an exemplary telecommunications session between a PSTN telephony device 68a and one or more multicast IP telephony devices 22, 23, 25, illustrated in Pearce Figure 1 below. (*See* Appx. G (Pearce), 10:55–61.) In Pearce, calls between IP telephony devices (e.g., devices 22, 23) and non-IP telephony devices 68 "are made through a gateway" (e.g., gateway 64). (Appx. G (Pearce), 5:9–12.) A gateway 64 "converts analog or digital circuit-switched data transmitted by PSTN 60 to packetized data transmitted by LAN 20, and vice-versa." (Appx. G (Pearce), 5:14–16.) Thus, the combination of Pearce and Oran discloses that "*at least one of said second subset of the plurality of clients is using a telephone*" [5]. (Appx. C (Bress Decl.), ¶164.) As discussed in Section V.A.1.a, gateway 64a in Pearce is a unicast device and therefore does not have the capability to mix audio during a telecommunications session. Therefore, the combination of Pearce and Oran also discloses "*at least one of the plurality of clients, which does not have the capability to mix multiple audio streams, is using a telephone*" [10]. (Appx. C (Bress Decl.), ¶164.)



Salama explains that "H.323 is a standard architecture for multimedia conferencing (voice, video, and[] data) in packet-based networks that was designed by the ITU-T. H.323 has been successfully applied as a suite of signaling protocols for Internet Telephony." (Appx. L (Salama), 2:9–13.) As depicted in Figure 13 of Salama below, a POSITA would have understood that the H.323 protocol could be used with a telephone. (Appx. C (Bress Decl.), ¶165.)

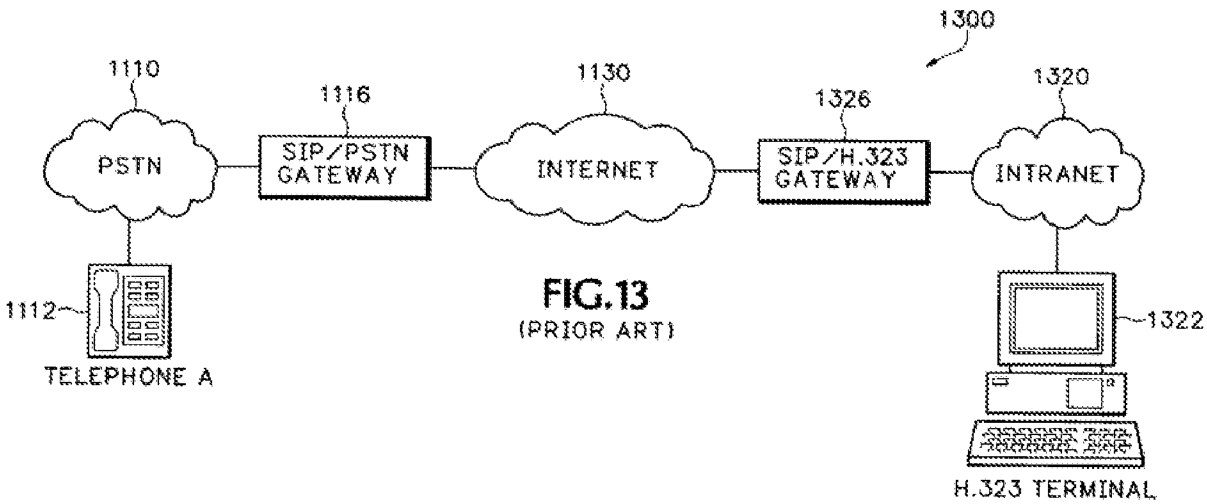


FIG.13
(PRIOR ART)

Accordingly, the combination of Pearce, Oran, and Salama, or Pearce, Oran, [Hoshi or Rosenberg], and Salama, discloses that "*at least one of said second subset of the plurality of clients is using a telephone and the H.323 protocol* [via a gateway]" [5] and that "*at least one of the plurality of clients, which does not have the capability to mix multiple audio streams, is using a telephone and the H.323 protocol* [via a gateway]" [10]. (Appx. C (Bress Decl.), ¶¶162–65.)

E. Ground X: Claims 1-3, 5-8, and 10 Are Obvious in View of Botzko in Combination with Kumar

1. Overview of the Combination

a. Overview of Botzko

The application which issued as United States Patent No. 6,141,597 to Botzko, et al ("Botzko"; Appx. M) was filed on September 8, 1997 and issued on October 31, 2000. Botzko is prior art under at least pre-AIA 35 U.S.C. § 102(e).

Botzko is directed to audio processors which are "responsible for receiving audio from various sites connected to the conference system and for distributing the audio to the various sites." (Appx. M (Botzko), 1:4–12.) Botzko's audio conferencing system, illustrated in Figure 1 below, includes a server (or bridge) 12 that receives "compressed audio, typically compressed audio packets" from a plurality of sites. (Appx. M (Botzko), 3:65–4:9.) Bridge 12 "operates to selectively forward, and/or mix, the audio from the various SITES so that each SITE can participate in a conference." (Appx. M (Botzko), 4:10–12.) Botzko notes that "[w]hile four SITES are shown in this illustrated embodiment, more or fewer SITES can be handled." (Appx. M (Botzko), 4:12–14.)

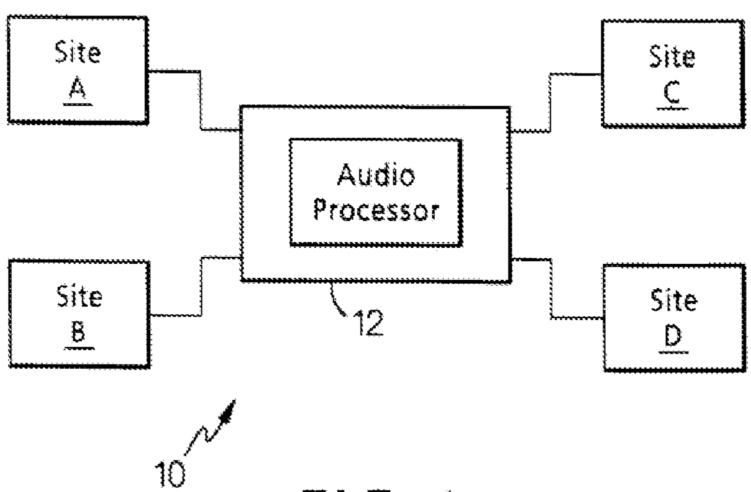


FIG. 1

Botzko, Figure 1

Botzko explains that "[t]here are two classic types of audio processors: an audio switch and an audio mixer." (Appx. M (Botzko), 1:13–14.) An "audio switch

is very efficient to implement because it is not required to decode the time compressed audio signals." (Appx. M (Botzko), 1:34–35.) An audio mixer, in contrast, "operates with non-time compressed, that is, uncompressed, audio." (Appx. M (Botzko), 1:56–57.) For each site in a conference, "the audio mixer combines the audio from selected other sites and re-encodes (that is, time compresses) the combined audio so that it can output time compressed, mixed audio to a receiving site." (Appx. M (Botzko), 1:56–61.) Botzko discloses both types of audio processors.

(i) Switching audio processor for local mixing clients

In Botzko, "[s]ome end-point SITES are able to receive more than one audio stream, and perform their own local mixing." (Appx. M (Botzko), 5:39–40.) Figure 2 (reproduced below) depicts an embodiment of Botzko's conferencing system including a switching audio processor capable of delivering multiple audio streams. As shown in Figure 2, "each one of the SITES 'A'-'D', transmits and receives time compressed audio packets, here, for example, through an RTP/RTCP transport." (Appx. M (Botzko), 4:34–36.) The compressed audio signals from the SITES "are fed to the audio processing section 15, of bridge 12 over lines 16a-16d [highlighted in red], respectively." (Appx. M (Botzko), 4:39–41.) The compressed audio signals "are also passed to time de-compressors, or decoders 18a-18d, respectively" which "produce corresponding de-compressed, or uncompressed,

audio signals on lines 19a-19d [highlighted in blue], respectively." (Appx. M (Botsko), 4:42–46.)

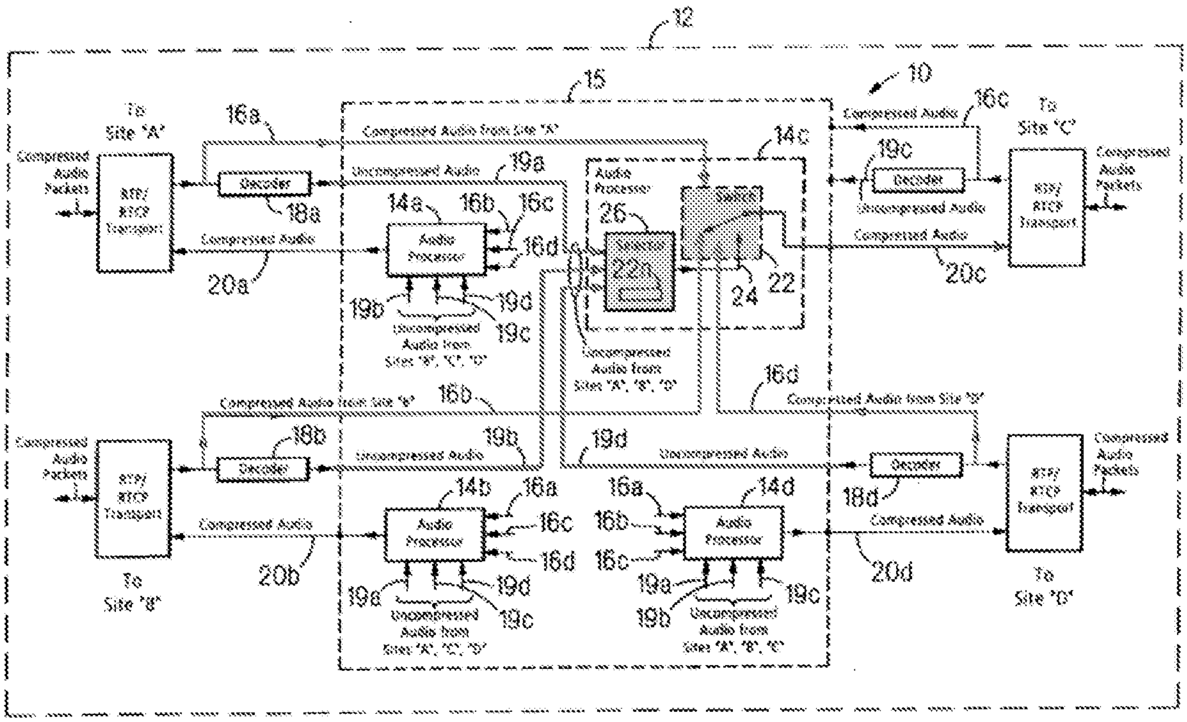


FIG. 2

Botzko, Annotated Figure 2

Audio processors 14a, 14b, 14c and 14d are each "coupled to a corresponding one of the sites SITE 'A', SITE 'B', SITE 'C', and SITE 'D', respectively, as shown, for example through RTP/RTCP transport circuits." (Appx. M (Botzko), 4:2–7.) An audio processor, e.g., audio processor 14c, "includes a switch 22 for receiving compressed audio signals" [highlighted in red] for each one of the plurality of SITES, other than the SITE served by the audio processor. (*See* Appx. M (Botzko), 4:53–57.) For example, switch 22 in audio processor 14c

receives compressed audio signals on lines 16a, 16b, and 16d "from each one of the plurality of SITES 'A', 'B' and 'D', respectively." (Appx. M (Botzko), 4:53–57.) Switch 22 "couples one of the plurality of compressed audio signals on lines 14a, 14b, 14d to SITE 'C', selectively, in accordance with and based upon a control signal on line 24" from selector 26. (Appx. M (Botzko), 4:57–60.)

Selector 26 (shaded blue) "is fed by the uncompressed audio signals on lines 19a, 19b, and 19d, from SITES 'A', 'B', and 'D', respectively", but not from line 19c for SITE 'C.' (Appx. M (Botzko), 4:60–62.) Selector 26 "determines the one of the SITES 'A', 'B', or 'D' with the highest likelihood of speech" and "produces the corresponding control signal on line 24" which couples the audio from one of SITES A, B, or D to SITE C. (Appx. M (Botzko), 4:62–5:1.) In an application where SITE C has the capability to "receive multiple audio streams, the selector 26 may be appropriately modified to select more than one of the SITES 'A', 'B', or 'D' for coupling to SITE 'C'." (Appx. M (Botzko), 5:1–4.)

Botzko explains that "there are at least three useful applications for the audio processors 14a-14d when they are communicating with end-point SITES." (Appx. M (Botzko), 6:4–6.) First, each SITE "unicasts its audio stream(s) to the conferencing bridge 12" which "selects, using selector 26, one or more streams at switches 22a, . . . , 22n, to unicast back to each end-point SITE." (Appx. M (Botzko), 6:6–10.) Second, each SITE "unicasts its audio stream(s) to the

conferencing bridge 12" which "selects, using selector 26, one or more streams at switches 22a, . . . , 22n, and multicasts the selected streams to all SITES." (Appx. M (Botzko), 6:10–14.) Third, each SITE "multicasts its audio stream(s)" which "can be received by the bridge 12, and also by other connected SITES." (Appx. M (Botzko), 6:14–16.) Bridge 12 then "selects one or more streams, and multicasts them on a separate multicast address." (Appx. M (Botzko), 6:16–18.)

(ii) Mixing audio processor for non-mixing clients

Botzko's platform also supports "normal end-points" which "can receive one audio stream to play out of their loudspeakers." (*See* Appx. M (Botzko), 5:22–25.) When multiple speakers are supported in a conference and one or more SITEs cannot mix, Botzko discloses alternative audio processor 14'c "which can be substituted for the audio processors" of Figure 2. (Appx. M (Botzko), 6:46–48.) Audio processor 14'c "operates as a mixer, and is adapted to operate with, for example, end-point SITES that can receive only one audio stream." (Appx. M (Botzko), 6:48–50.)

Selector 26' receives the "uncompressed audio signals on lines 19a, 19b, and 19d, decoded from the compressed signals from SITES 'A', 'B', and 'D', respectively." (Appx. M (Botzko), 6:62–65.) Using these signals, selector 26' "determines whether more than person is speaking at those sites at the same time." (Appx. M (Botzko), 7:20–23.) If more than one person is speaking at the same

time, "a logic '1' signal is fed to line 36; otherwise the selector 26' produces a logic '0' signal." (Appx. M (Botzko), 7:23–26.) When the logic signal on line 36 is logic '1', "the mixer 28 and encoder 29 are enabled." (Appx. M (Botzko), 7:28–33.) The detector also "provides to the mixer 28, over lines 37, signal information identifying on which of the input lines the speakers can be found." (Appx. M (Botzko), 7:54–57.) The uncompressed audio is then selectively mixed by "the mixer 28 and then is encoded, that is, time compressed, in encoder 29." (Appx. M (Botzko), 7:40–45.) The compressed composite audio signal is then fed to input 30 of selector 34. (Appx. M (Botzko), 7:11–13.)

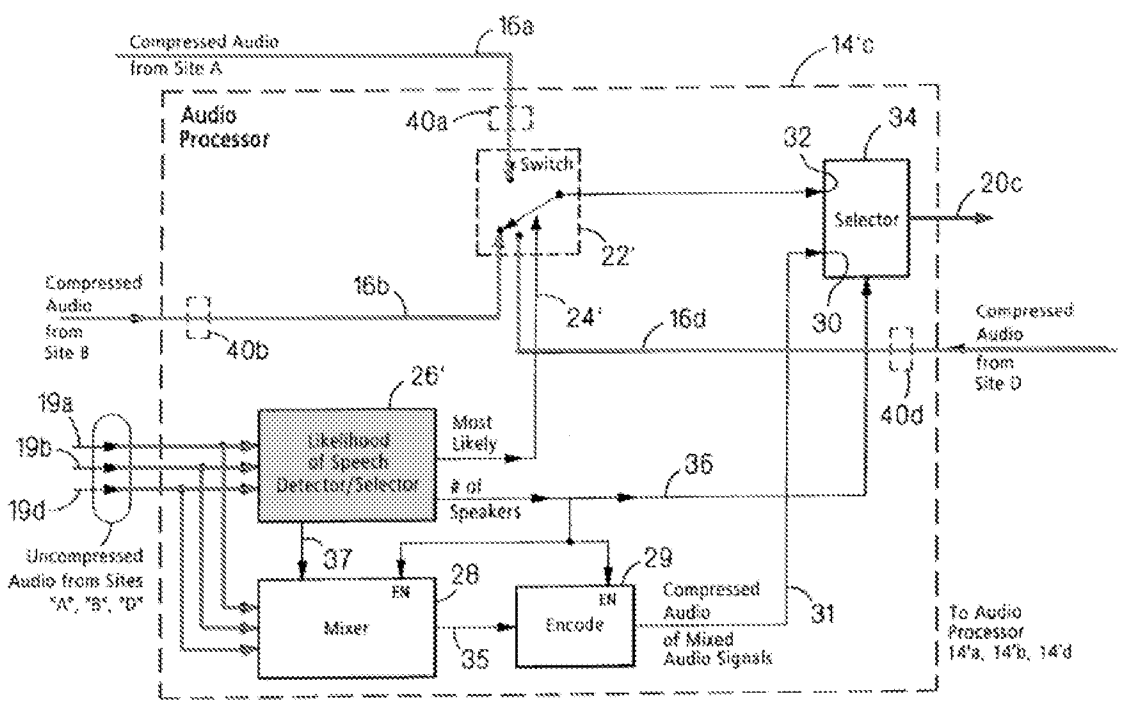


FIG. 3

Botzko, Annotated Figure 3

Selector 26' also "determines the one of the SITES 'A', 'B', or 'D' with the highest (or loudest) likelihood of speech to produce the control signal on line 24'." (Appx. M (Botzko), 6:65–7:2.) Switch 22' in turn "couples one of the plurality of compressed audio signals on lines 16a, 16b, 16d to input 32 of selector (or switch) 34, selectively, in accordance with a control signal on line 24'." (Appx. M (Botzko), 6:59–62.) Selector 34 is therefore "fed at one input 32 with the one of the compressed audio signals from SITES 'A', 'B', and 'D', having the most likely (or loudest) speaker and the other one of the inputs 30 is fed with the time compressed composite (mixed) audio signal produced by the encoder 29." (Appx. M (Botzko), 7:14–19.) When "more than one person is speaking at the same time, the mixer 28 and encoder 29 are enabled and selector 34 couples the time compressed composite audio signal produced by the encoder 29 through the selector 34 to the SITE 'C'." (Appx. M (Botzko), 7:29–33.) Otherwise, when only one person is speaking, "selector 34 couples the selected one of the compressed audio signals on lines 16a, 16b, 16d, having the most likelihood of speech, through the selector 34 to the SITE 'C', over line 20c." (Appx. M (Botzko), 7:33–39.)

(iii) Centralized Components

Botzko illustrates an alternative embodiment of its bridge having a single selector in Figure 3A (reproduced below). In this embodiment, up to three speakers can be identified and mixed together. (Appx. M (Botzko), 8:28–30.) Based on

Botzko's teachings, a POSITA would have understood that additional mixers and encoders would be added if additional speakers are necessary. (Appx. C (Bress Decl.), ¶175; *See, e.g.,* Appx. M (Botzko), 8:54–56 ("In the preferred embodiment of the invention, the mixer/encoders and selector, as well as the aligners and decoders, are all implemented in software").) In this embodiment, "compressed output of the aligners are sent to each of four mixers 28a, 28b, 28c, and 28d and the uncompressed audio output [indicated by blue lines] is sent to a selector 26'." (Appx. M (Botzko), 8:35–37.) The selector controls each of the mixers (shaded green) "over lines 37 and each mixer, when enabled, produces a mixed output to an encoder 29a, 29b, 29c, 29d, respectively, which generate appropriate output signals for the SITES A, B, C, and D" [highlighted in green]. (Appx. M (Botzko), 8:38–42.) The encoder outputs are "directed to a cross-point switch 100." (Appx. M (Botzko), 8:42–43.)

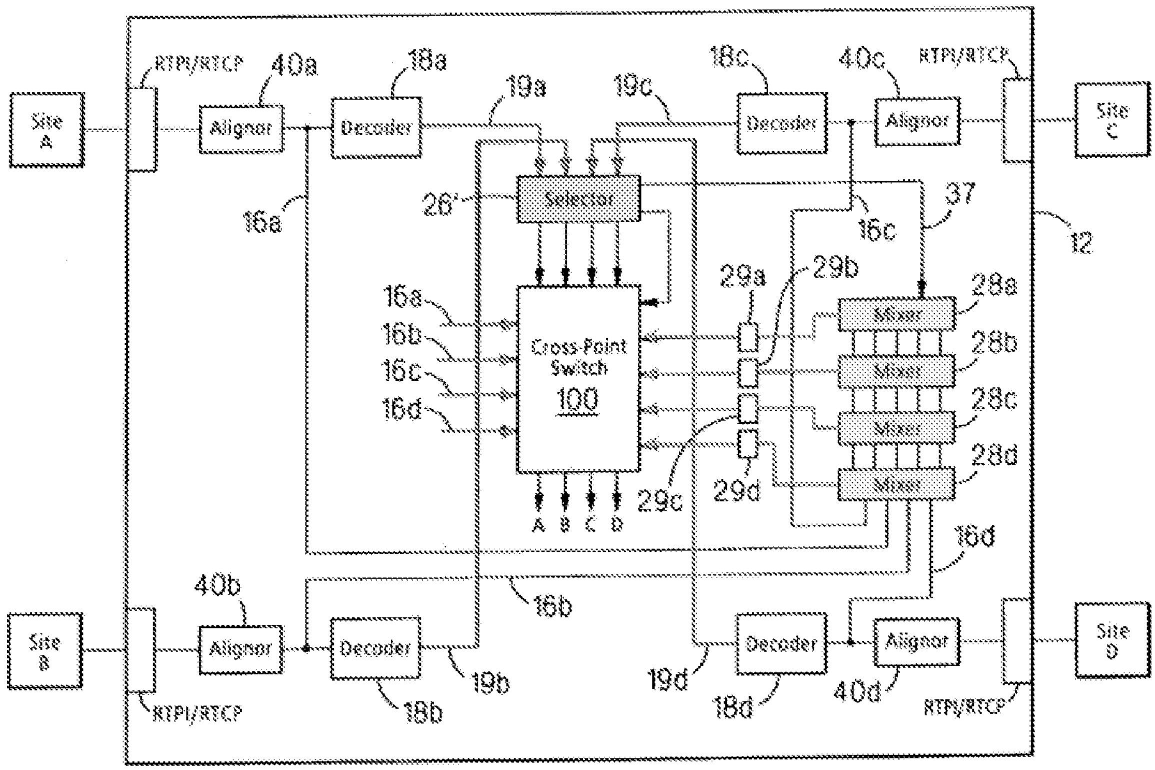


FIG. 3A

Botzko, Annotated Figure 3A

Cross-point switch 100 "also receives the compressed inputs from the SITES, output by the aligners, over lines 16a, 16b, 16c, 16d." (Appx. M (Botzko), 8:43–48.) The selector output "controls the cross-point switch to select either the compressed audio over lines 16 or the outputs of the encoders 29 for presentation to the various SITES over lines labeled A, B, C, and D." (Appx. M (Botzko), 8:48–52.)

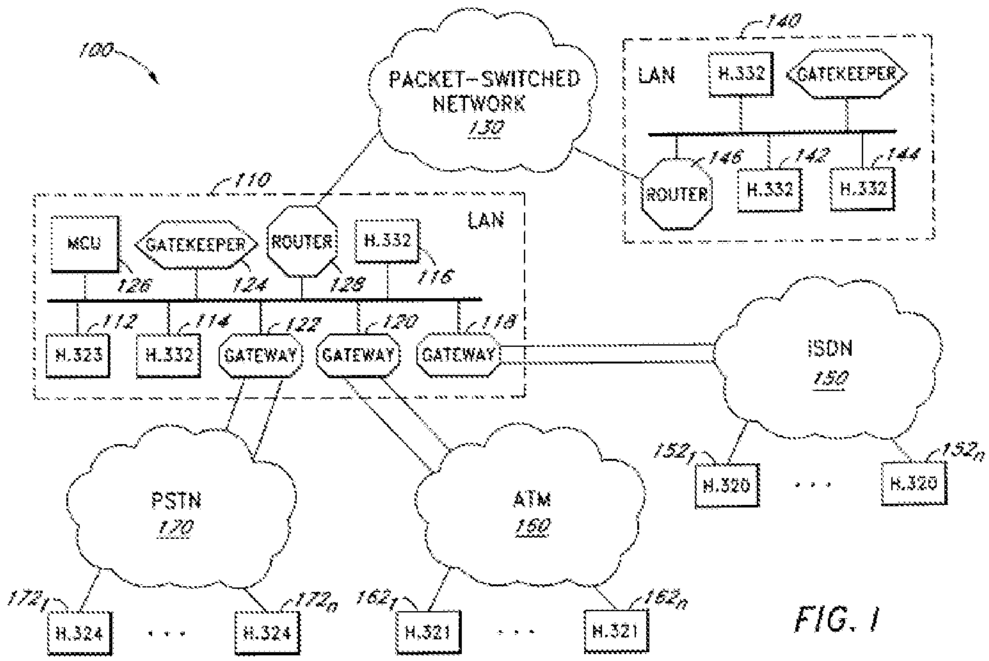
Botzko's bridge 12 simultaneously supports both switching audio processors, to send one or more audio streams of selected speakers to SITEs that are capable of mixing multiple audio streams, and mixing audio processors, to send

one audio stream with audio from selected speakers mixed to SITEs that are not capable of mixing multiple audio streams. (*See e.g.*, Appx. M (Botzko) 4:10–12 (Bridge 12 "operates to selectively **forward, <u>and/or</u> mix**, the audio from the various SITES so that each SITE can participate in a conference").) A POSITA would understand that the identification by Selector 26 of the active speakers can be used to control cross-point switch to couple one or more streams 16a-163 to one or more SITES A-D in the manner disclosed in Figures 2, 2A, and 2B or to couple mixed audio for 1, 2, or 3 speakers to one or more SITEs. (Appx. C (Bress Decl.), ¶177.)

b. Overview of Kumar

Kumar "relates generally to the field of multimedia communication, and specifically, to a method and apparatus to provide a back-channel for receiver terminals in a loosely-coupled conference." (Appx. N (Kumar), 1:9–12.) Kumar's system 100, illustrated in Figure 1 below, "includes a plurality of terminals (e.g., H.323 terminal 112 and H.332 terminals 114 and 116) coupled together in a network 110 such as a local area network ('LAN')." (Appx. N (Kumar), 3:28–31.) A multipoint control unit ("MCU") includes a multipoint controller ("MC") and "zero to more multipoint processors ('MPs')." (Appx. N (Kumar), 3:45–49.) A MC "provides control functions for a conference while the MP receives media streams

from the terminals, processes the media streams, and returns them to the terminals in the conference." (Appx. N (Kumar), 3:49–53.)

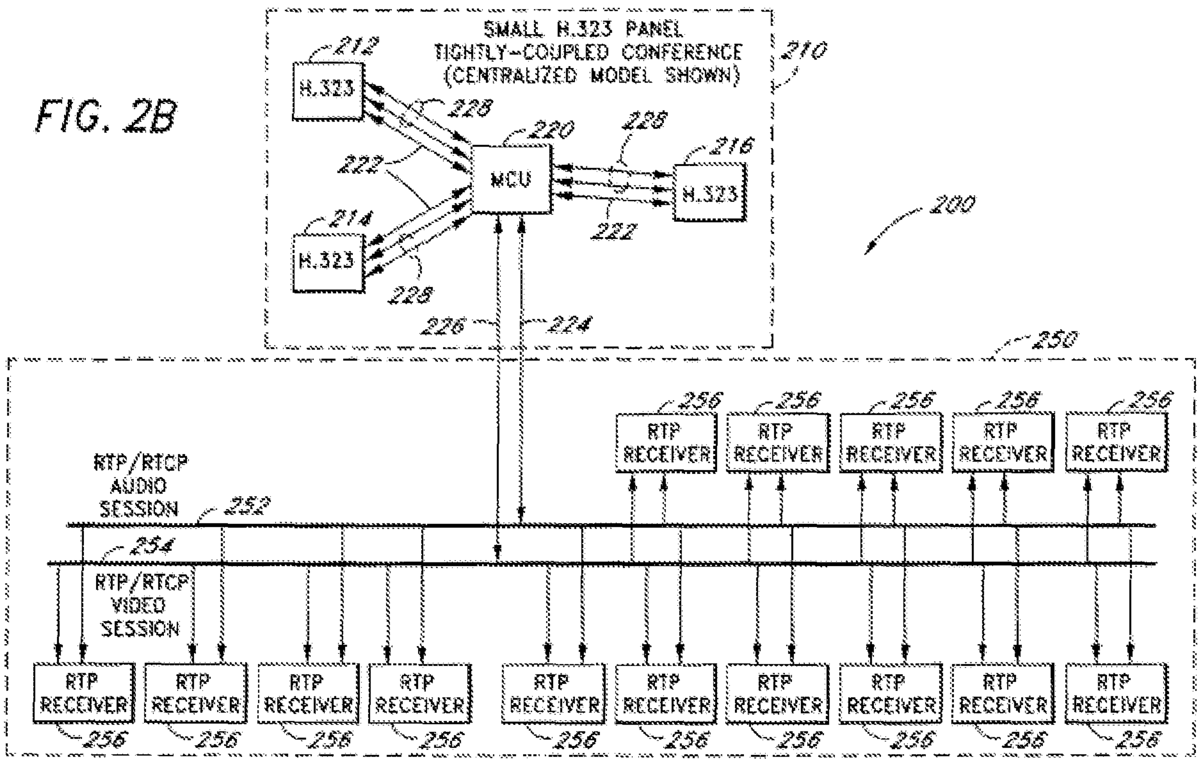


Kumar, Figure 1

A centralized model of Kumar's system is illustrated in Figure 2B. The system 200 includes a small H.323 panel 210 and a large group 250 of RTP receiver terminals 256. "Within the panel 210, full interaction is allowed between terminals 212, 214, 216, and 218 either through social or automatic control." (Appx. N (Kumar), 3:63–64.) In social control, every participants "in the panel can potentially talk and transmit their audio and video on the RTP/RTCP audio and video sessions 252 and 254." (Appx. N (Kumar), 3:65–67.) The panel terminals 212, 214, and 216 "communicate with a MC of the MCU in a point-to-point

manner on the H.245 control channel and also with a MP of the MCU 220 in a point-to-point manner on the audio and video channels 228." (Appx. N (Kumar), 4:33–37.) RTP receiver terminals 256, in contrast, "are passive and are not allowed to interact with the conference." (Appx. N (Kumar), 4:53–54.)

FIG. 2B



Kumar, Figure 2B

In Kumar's method, after a conference is pre-announced, "capability negotiation is performed." (Appx. N (Kumar), 5:63–64.) The capability negotiation server "may also be the MC of the announced conference." (Appx. N (Kumar), 8:12–13.) Kumar teaches that "[i]n the preferred embodiment, the procedures outlined in H.323 are followed." (Appx. N (Kumar, 8:14–15.) The H.323 standard

(incorporated by reference by Kumar) explains that capabilities exchange "shall follow the procedures of Recommendation H.245, which provides for separate receive and transmit capabilities." (Appx. P (H.323 11/96), 17) The receive capabilities "describe the terminal's ability to receive and process information streams." (Appx. P (H.323 11/96), 17.) For example, the H.323 standard requires that the H.323 terminal "shall use H.245 simultaneous capabilities to indicate how many simultaneous audio streams it is capable of decoding." (Appx. P (H.323 11/96, 13.)

c. Motivation to Combine

A POSITA would have been motivated to combine Botzko's conferencing system with Kumar's teachings of H.323 conference system. (*See generally* Appx. C (Bress Decl.), ¶¶181–85.) For example, a POSITA would have been motivated to integrate Kumar's MC and capabilities exchange/negotiation into Botzko's conferencing system.

Botzko's conferencing system supports normal end-point SITES "which can receive **one audio stream** to play out of their loudspeakers" (Appx. M (Botzko), 5:22–25) and SITES that "are able to receive **more than one audio stream**, and perform their own local mixing" (Appx. M (Botzko), 5:39–40). Therefore, to set-up a conference call, Botzko's conferencing system must determine which types of audio processors to couple to each SITE (e.g., mixing audio processor or switching

audio processor) and determine the number of active speakers that are supported for the call. (Appx. C (Bress Decl.), ¶182.) Botzko however does not disclose a technique for determining the local mixing capabilities of its SITES. Botzko does suggest that its bridge obtains capabilities information for the SITES. For example, Botzko teaches, e.g., that in its mixing audio processor, the "level of mixing will depend **upon the bridge configuration**, including, in particular, the number of connected SITES and the desirability of hearing more than two or three speakers at the same time." (Appx. M (Botzko), 7:69–63.) And for its switching audio processor, Botzko teaches that the "system can limit the number of streams it outputs to the number a SITE can receive." (Appx. M (Botzko), 5:56–58.) Based on Botzko's suggestions of utilizing SITE capability information to configure its conference bridge for a conference, a POSITA would have been motivated to look for references describing techniques for obtaining and determining capability information (including mixing capabilities) and would have been led to Kumar. (Appx. C (Bress Decl.), ¶182.)

Additionally, a POSITA would have been motivated to integrate Kumar's teachings related to H.323 conferences (e.g., MC and capabilities exchange) into Botzko to support H.323 conferences. (Appx. C (Bress Decl.), ¶183.) Because H.323 was a well-established standard by the filing date of the '858 patent, a POSITA would have been motivated to use Kumar's H.323 teachings to be able to

support a wider set of SITES and to expand the useability of Botzko's conference bridge to industry "standard" conferences. (Appx. C (Bress Decl.), ¶183.) A POSITA would have had a reasonable expectation of success in the combination because Kumar incorporates industry standard processes and because components of Botzko's bridge (e.g., mixer/encoders, selector, aligners, decoder) are implemented in software. (*See, e.g.,* Appx. M (Botzko), 8:54–57; Appx. C (Bress Decl.), ¶183.) It would have been within the capabilities of a POSITA by the filing date of the '858 patent to modify Botzko's software to incorporate the teachings of Kumar (and the H.323 standard). (Appx. C (Bress Decl.), ¶183.)

The proposed combination merely applies a known technique (Kumar's MC and capabilities exchange) to a known device (Botzko's platform 12) ready for improvement to yield predictable results. (Appx. C (Bress Decl.), ¶184.) As discussed above, the results would be predictable because Kumar is describing the operation of an established industry standard and the components of Botzko's bridge are software-based. (Appx. C (Bress Decl.), ¶184.)

Finally, as described above, Botzko makes numerous references to supporting SITEs that have the capability to mix multiple received audio streams, and SITEs that are capable of only receiving one audio stream and therefore unable to mix audio. Given that H.323 was a relevant telecommunications industry standard for teleconferencing before and at the time of the '858 patent, it would

have therefore been obvious to a POSITA to incorporate the "capabilities exchange" method disclosed in the H.323 standard into Botzko's conferencing bridge system. (Appx. C (Bress Decl.), ¶185.)

2. The combination of Botzko and Kumar renders independent claim 1 obvious.

a. The combination of Botzko and Kumar discloses the preamble.

To the extent the preamble is limiting, Botzko discloses a "*method of providing audio conferencing for a plurality of clients using varying equipment and protocols*" [1P]. (Appx. C (Bress Decl.), ¶¶186–89.) Botzko is directed to audio processors which are "responsible for receiving audio from various sites connected to the conference system and for distributing the audio to the various sites." (Appx. M (Botzko), 1:4–12.) Botzko's "audio conferencing system", illustrated in Figure 1 below, includes a plurality of SITES (e.g., SITES A, B, C, and D) "connected together through a server, or bridge 12." (Appx. M (Botzko), 3:65-4:2.) Bridge 12 "operates to selectively forward, and/or mix, the audio from the various SITES so that each SITE can participate in a conference." (Appx. M (Botzko), 4:10-12.) Botzko's SITES are the recited "*plurality of clients*." (Appx. C (Bress Decl.), ¶187.)

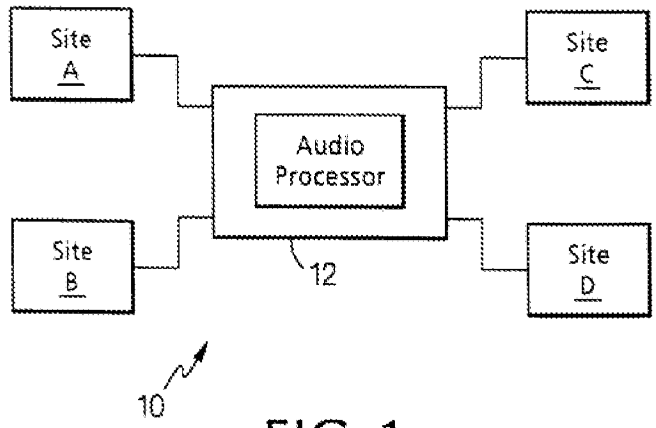


FIG. 1

Botzko, Figure 1

Botzko's SITES "*us[e] varying equipment and protocols*." (Appx. C (Bress Decl.), ¶¶188–89.) Botzko teaches that SITES connected to the bridge include "normal end-points, which can receive one audio stream to play out of their loudspeakers." (Appx. M (Botzko), 5:22–25.) Thus, the equipment at these devices can handle only a single stream and therefore do not have the capability to mix audio streams for a conference. (Appx. C (Bress Decl.), ¶188.) Botzko's system also supports "end-point SITES [that] are able to receive more than one audio stream, and perform their own local mixing." (Appx. M (Botzko), 5:39-40.) The equipment at these SITES handles multiple streams and has the capability to mix. (Appx. C (Bress Decl.), ¶188.) Thus, Botzko's SITES use varying equipment. (Appx. C (Bress Decl.), ¶188.)

In addition, Botzko's SITES "can connect to the bridge in many different ways, for example, through the public switched telephone network, by wireless, by

direct connection, or in any other desired combination of the various communication paths including, for example, a local area network.” (Appx. M (Botzko), 4:18-24.) A POSITA would understand that a client connected to the PSTN, a client connected to a wireless network, and a client connected to a LAN would use varying protocols for communications with the network. (Appx. C (Bress Decl.), ¶189.)

Thus, Botzko discloses “*providing audio conferencing for a plurality of clients using varying equipment and protocols*” [1P]. (Appx. C (Bress Decl.), ¶¶186–89.)

b. The combination of Botzko and Kumar discloses “receiving an audio packet from each of the plurality of clients” [1.1].

Botzko discloses “*(1) receiving an audio packet from each of the plurality of clients*” [1.1]. (Appx C (Bress Decl.), ¶¶190–91.) In Botzko, “each one of the SITES ‘A’ - ‘D’, **transmits** and receives **time compressed audio packets**, here, for example, through an RTP/RTCP transport.” (Appx. M (Botzko), 4:34–36; Figure 2 (below).) “The **audio data received from** and sent to the various remote SITES A, B, C, and D from the bridge 12 is compressed audio, **typically compressed audio packets**.” (Appx. M (Botzko), 4:7–9.) The time compressed audio signals from audio sources at the SITES “are fed to the audio processing section 15, of bridge 12 over lines 16a-16d, respectively.” (Appx. M (Botzko), 4:39-41.)

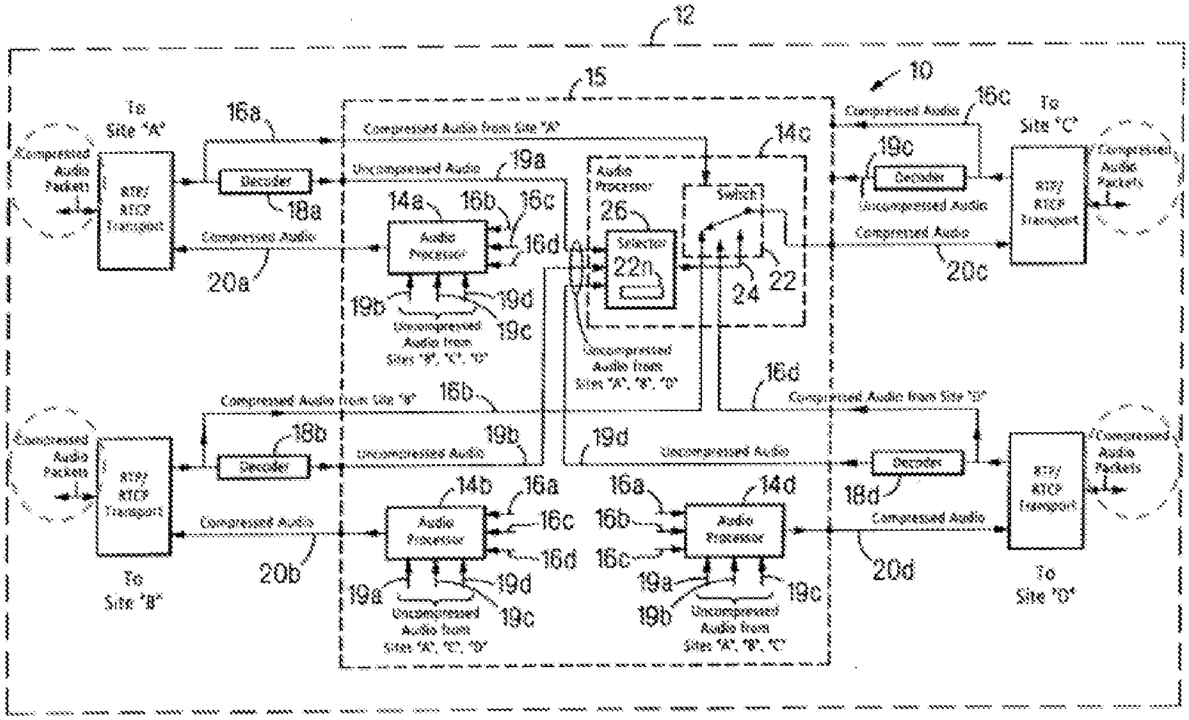


FIG. 2

Botzko, Annotated Figure 2

c. The combination of Botzko and Kumar teaches "(2) determining which of the plurality of clients is an active speaker … " [1.2].

The combination of Botzko and Kumar teaches "*(2) determining which of the plurality of clients is an active speaker and forming an active speakers list*" [1.2]. (Appx. C (Bress Decl.), ¶¶192–98.)

Botzko describes operation of selector 26 is three contexts: (1) in switching audio processors for local mixing clients; (2) in mixing audio processors for non-mixing clients; and (3) in a platform combining switching and mixing. Botzko does not limit the number of "*active speakers*" that can be identified by its platform.

In a switching audio processor for local mixing clients (illustrated in annotated Figure 2 below), selector 26 "is fed by the uncompressed audio signals on lines 19a, 19b, and 19d, from SITES 'A', 'B', and 'D', respectively." (Appx. M (Botzko), 4:60-62.) Selector 26 "determines the one of the SITES 'A', 'B', or 'D' with the highest likelihood of speech" and "produces the corresponding control signal on line 24" which couples one of SITES A, B, or D to SITE C. (Appx. M (Botzko), 4:62-5:1.) When a SITE (e.g., SITE 'C') has the capability to receive multiple audio streams, "the selector 26 may be appropriately modified to select more than one of the SITES" (e.g., SITES 'A', 'B', or 'D') for coupling to its end-point SITE (e.g., SITE 'C'). (Appx. M (Botzko), 5:1-4.) For these SITES, "a more complex selector may be required." (Appx. M (Botzko), 5:45-46.) Botzko's selector 26 therefore "*determine[es] which of the plurality of clients is an active speaker*." (Appx. C (Bress Decl.), ¶194.)

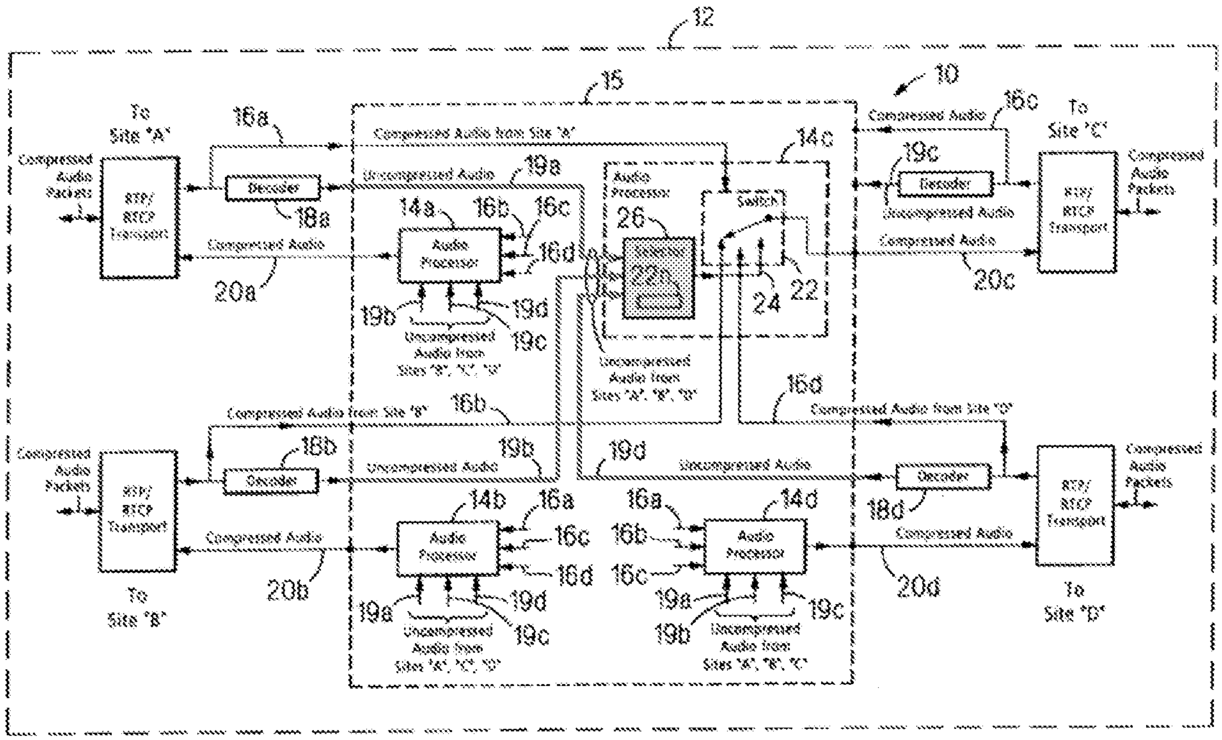


FIG. 2

Botzko, Annotated Figure 2

In a mixing audio processor for non-mixing clients (illustrated in annotated Figure 3 below), selector 26' (shaded blue) "connects to the uncompressed audio signals on lines 19a, 19b, and 19d, decoded from the compressed signals from SITES 'A', 'B', and 'D', respectively." (Appx. M (Botzko), 6:62-65.) Selector 26' "includes a likelihood of speech detector and determines the one of the SITES" having "the highest (or loudest) likelihood of speech." (Appx. M (Botzko), 6:65-7:2.) In addition, selector 26' "determines whether more than person is speaking at those sites at the same time." (Appx. M (Botzko), 7:20-23.) "When two or more speakers are determined to exist by selector 26' the detector provides to the mixer

28, over lines 37, signal information identifying on which of the input lines the speakers can be found." (Appx. M (Botzko), 7:54-57.) Botzko's selector 26' therefore also "*determine[es] which of the plurality of clients is an active speaker*." (Appx. C (Bress Decl.), ¶195.)

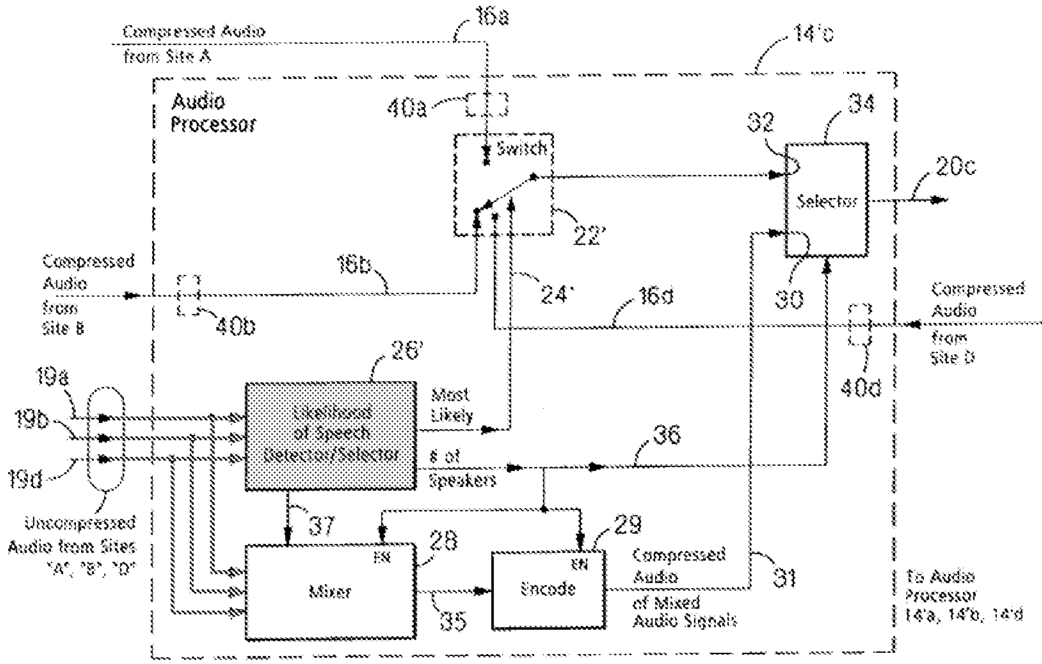


FIG. 3

Botzko, Annotated Figure 3

Botzko illustrates an alternative embodiment having a single selector in Figure 3A (reproduced below). In this embodiment, up to three speakers can be identified and mixed together. (Appx. M (Botzko), 8:28-30.) Based on Botzko's teachings, a POSITA would have understood that additional mixers and encoders would be added if additional speakers are necessary. (Appx. C (Bress Decl.), ¶175; *See, e.g.,* Appx. M (Botzko), 8:54-56 ("In the preferred embodiment of the

invention, the mixer/encoders and selector, as well as the aligners and decoders, are all implemented in software").) In this embodiment, "compressed output of the aligners are sent to each of four mixers 28a, 28b, 28c, and 28d and the uncompressed audio output [indicated by blue line] is sent to a selector 26'." (Appx. M (Botzko), 8:35-37.)

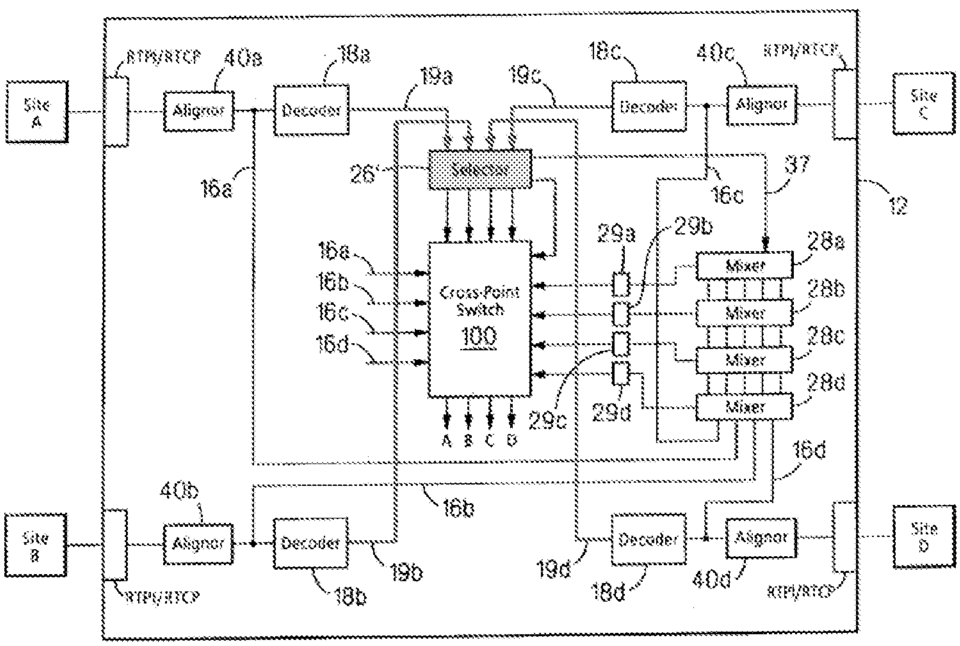


FIG. 3A

Botzko, Annotated Figure 3A

Botzko's selector 26 and selector 26' each determines "*which of the clients is an active speaker[s]*." (Appx. C (Bress Decl., ¶197.) Botzko notes that "selected speakers will also depend upon, and typically be selected from SITES having, **a certain minimum threshold level of speech**." (Appx. M (Botzko), 7:63-66.) Botzko further teaches that "an acoustic energy detector, or other device could be

used as equivalent alternatives for the selector 26." (Appx. M (Botzko), 5:4-6.) While Botzko uses the term "loudest speakers", Botzko equates "loudest speakers" to "active speakers: "Each bridge could then multicast the 'active' audio streams to a multicast group 96." (Appx. M (Botzko), 6:36-38; *see also*, 6:31-34 ("The ability of the audio processors 14a-14d to select those audio streams which contain speech from the totality of streams, without degrading the audio, is very useful in distributed conferences").) From these disclosures, a POSITA would therefore understand that Botzko teaches that its selector determines which SITES "*is an active speaker*." (Appx. C (Bress Decl.), ¶197.)

Botzko therefore discloses "*(2) determining which of the plurality of clients is an active speaker*." (Appx. C (Bress Decl.), ¶¶192–97.)

Botzko's selector, as discussed above, determines which of the SITES involved in a conference are "active speakers." Although Botzko does not explicitly disclose the formation of an "*active speaker list*", it would have been obvious to a POSITA, based on the teachings of Botzko, that the "active speaker" information is stored in a list. (Appx. C (Bress Decl.), ¶198.) Botzko discloses that its selector "provides to the mixer, over lines 37, **signal information** identifying on which of the input lines **the speakers can be found**." (*See* Appx. M (Botzko), 7:54-57.) Botzko's selector also configures switches to select the appropriate audio streams for delivery to local mixing clients. (*See, e.g.,* 5:39-51; Figures 2A and

2B). Based on these disclosures, a POSITA would understand that Botzko stores the identification of the active speakers so that the Selector(s) can instruct the mixer and/or switches of the appropriate signal lines to mix (for non-mixing SITES) or connect to the RTP/RTCP transport circuit (for local mixing SITES). (Appx. C (Bress Decl.), ¶198.)

Further, Botzko teaches that "the mixer/encoders and selector, as well as the aligners and decoders, are all implemented in software." (Appx. M (Botzko), 8:54-56).) A POSITA would therefore have understood that for "**signal information** identifying on which of the input lines **the speakers can be found"** to be sent over "line 37" from the selector to the mixer (as shown in Botzko Fig. 3) in software, a list of active speakers would be required to convey that information. (Appx. C (Bress Decl.), ¶198.)

A POSITA would understand that the identified active speakers in the combination of Botzko and Kumar would be stored as a list. (Appx. C (Bress Decl.), ¶198.) A list is merely a "multi-element data structure that has a linear (first, second, third, …) organization but that allows elements to be added or removed in any order." (Appx. E (Microsoft Dictionary), 270.) First, a list is one of the most natural and oldest forms of organizing data. (Appx. C (Bress Decl.), ¶198.) Second, even prior to the '858 patent, lists were one of the very first data structures taught in programming. (Appx. C (Bress Decl.), ¶198.)

Accordingly, the combination of Botzko and Kumar teaches or at least suggests "*forming an active speakers list*." (Appx. C (Bress Decl.), ¶198.)

d. The combination of Botzko and Kumar discloses "(3) determining that a first subset of the plurality of clients has the capability to mix …" [1.3] and "(4) determining that a second subset of the plurality of clients does not have the capability to mix …" [1.4].

The combination of Botzko and Kumar discloses "*(3) determining that a first subset of the plurality of clients has the capability to mix multiple audio streams*" [1.3] and "*(4) determining that a second subset of the plurality of clients does not have the capability to mix multiple audio streams*" [1.4]. (Appx. C (Bress Decl.), ¶¶200–02.)

Bridge 12 "operates to selectively forward, and/or mix, the audio from the various SITES so that each SITE can participate in a conference." (Appx. M (Botzko), 4:10-12.) Botzko's conferencing system also supports multiple types of SITES: normal end-point SITES "which can receive **one audio stream** to play out of their loudspeakers" (Appx. M (Botzko), 5:22-25) and SITES that "are able to receive **more than one audio stream**, and perform their own local mixing" (Appx. M (Botzko), 5:39-40). A POSITA would understand that a SITE that can only receive a single audio stream "*does not have the capability to mix*." (Appx. C (Bress Decl.), ¶201.) Thus, Botzko's conferencing system makes processing decisions, e.g., the type of audio processor to couple to a SITE and whether to mix

155853739.8
155853739.12

or forward multiple streams, based on the capabilities of the SITES but Botzko does not disclose a technique for obtaining the mixing capabilities of its SITES. Kumar provides this disclosure.

Kumar discloses an MC that "provides control functions for a conference." (Appx. N (Kumar), 4:49-53.) In Kumar, terminals "communicate with a MC of the MCU in a point-to-point manner on the H.245 control channel." (Appx. N (Kumar), 4:33-37.) Specifically, as set forth in H.323 (which is incorporated by reference in Kumar), requires that "[e]ndpoint system capabilities are exchanged by transmission of the H.245 **terminalCapabilitySet** message" after call set-up messages are exchanged. (Appx. P (H.323 11/96), 48 (emphasis in original).) The H.323 standard notes that an H.323 terminal "may need to perform an audio mixing function in order to present a composite audio signal to the user." (Appx. P (H.323 11/96), 13.) Accordingly, for "audio mixing," H.323 requires that H.323 terminals "shall use H.245 simultaneous capabilities to indicate how many simultaneous audio streams it is capable of decoding." (Appx. P (H.323 11/96, 13.) If the "audio mixing' capability received for a SITE indicates that the terminal can only decode one stream, a POSITA would understand that the terminal is not capable of performing local mixing. (Appx. C (Bress Decl.), ¶202.) And, if the "audio mixing" capability received for a SITE indicates that the terminal can decode two or more streams, a POSITA would understand that the terminal is

capable of performing local mixing. (Appx. C (Bress Decl.), ¶202.) Thus, in the combination, the MC associated with Botzko's bridge determines the mixing capability of the SITES using Kumar's H.323's capabilities exchange and sets up the configuration of the bridge for the conference. (Appx. C (Bress Decl.), ¶202.)

Accordingly, the combination of Botzko and Kumar discloses "*(3) determining that a first subset of the plurality of clients has the capability to mix multiple audio streams*" [1.3] and "*(4) determining that a second subset of the plurality of clients does not have the capability to mix multiple audio streams*" [1.4]. (Appx. C (Bress Decl.), ¶¶200–02.)

e. The combination of Botzko and Kumar discloses or at least suggests "(5) multiplexing said packets of audio data received from each client on said active speakers list into a multiplexed stream" [1.5].

The combination of Botzko and Kumar discloses or at least suggests "*(5) multiplexing said packets of audio data received from each client on said active speakers list into a multiplexed stream*" [1.5]. (Appx. C (Bress Decl.), ¶¶204–10.)

Botzko explains that "[t]here are two classic types of audio processors: an audio switch and an audio mixer." (Appx. M (Botzko), 1:13-14.) Botzko's Figure 2 illustrates a switching audio processor that provides a single audio stream to a SITE. (*See, e.g.*, Overview of Botzko.) However, Botzko also teaches that "[s]ome end-point SITES are able to receive more than one audio stream, and perform their own local mixing." (Appx. M (Botzko), 5:39-40.) To support these SITES, "extra

switches 22 and a more complex selector may be required." (Appx. M (Botzko), 5:45-46.)

Botzko describes two applications using multiple output streams. In a first, application "each end-point SITE unicasts its audio stream(s) to the conferencing bridge 12" which "selects, using selector 26, one or more streams at switches 22a, . . . , 22n, to unicast back to each end-point SITE. (See FIG. 2A)." (Appx. M (Botzko), 6:6-10; Figure 2A (below).) Botzko's audio processors 14 are "coupled to a corresponding one of the sites" (e.g., SITE 'A', SITE 'B', SITE 'C', and SITE 'D') "through RTP/RTCP transport circuits." (Appx. M (Botzko), 4:2-7.) In this application, audio processor 14 receives compressed audio from SITES (e.g., SITES A, B, and D) other than the SITE it is coupled to (e.g., SITE C). (*See* Appx. M (Botzko), Figure 2; Appx. C (Bress Decl.), ¶207.) Thus, this application does not include a SITE's own audio in the one or more audio streams switched and sent to the SITE (e.g., SITE 'C').

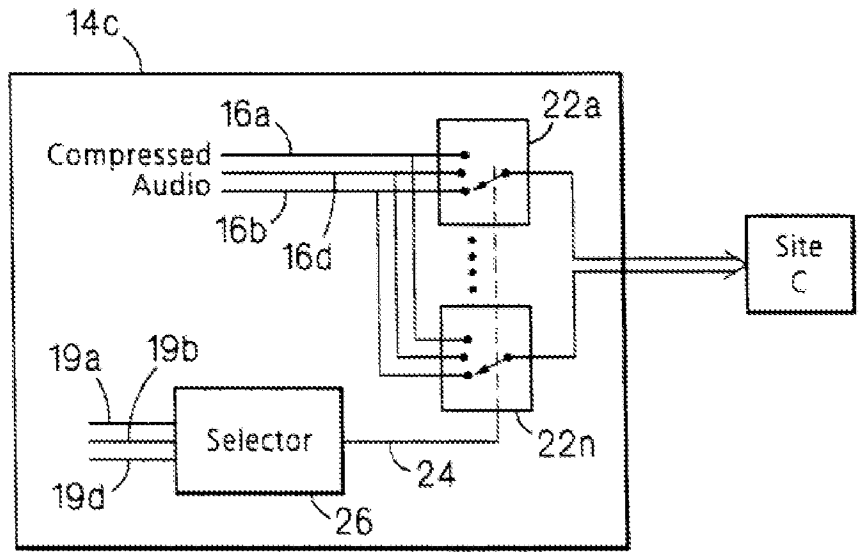


FIG. 2A

Botzko, Figure 2A

As shown in Botzko's Figures 2 (below), the output of audio processor 14c (multiple audio streams) [highlighted in purple] is sent as compressed audio to its corresponding RTP/RTCP transport circuit [shaded purple] in bridge 12. (*See* Appx. C (Bress Decl.), ¶207.)

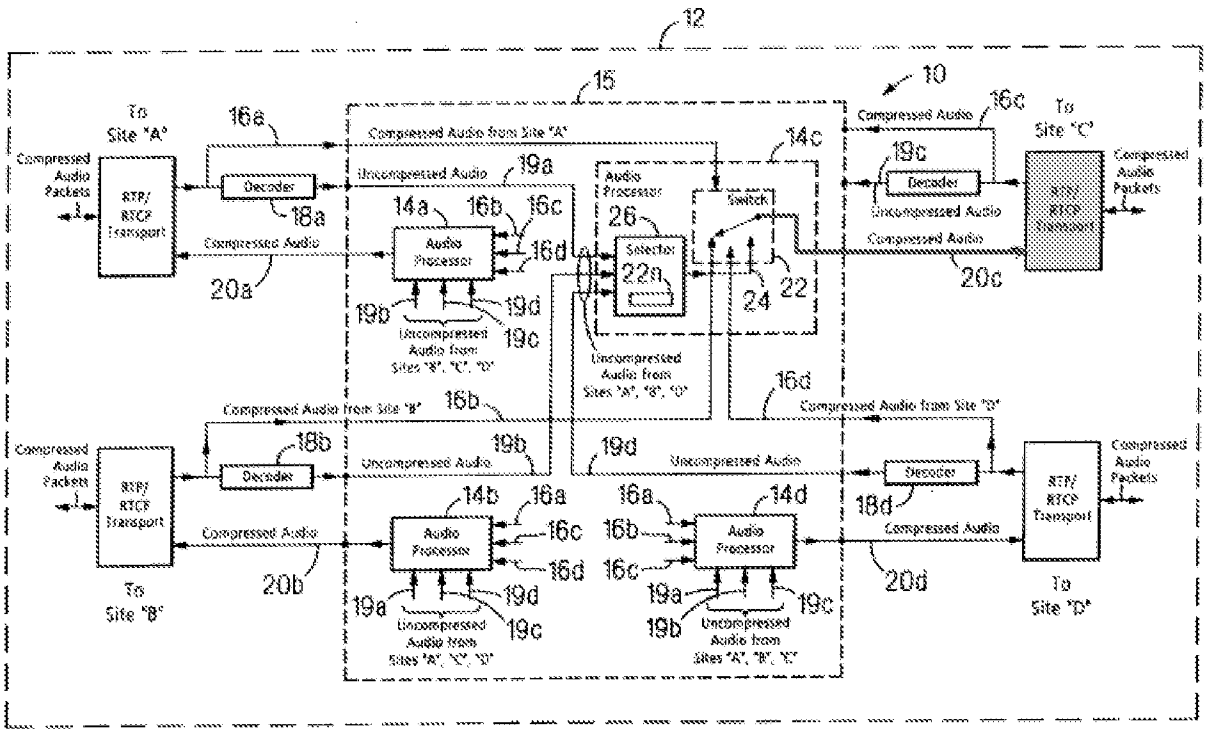


FIG. 2

Botzko, Annotated Figure 2

In a second application, "each end-point SITE unicasts its audio stream(s) to the conferencing bridge 12" which "selects, using selector 26, one or more streams at switches 22a, . . . , 22n, and multicasts the selected streams to all SITES." (Appx. M (Botzko), 6:10-14.) In this application, audio processor 14 receives compressed audio from all SITES (e.g., SITES A, B, C, and D), including the SITE it is coupled to (e.g., SITE C). (*See* Appx. M (Botzko), Figure 2; Appx. C (Bress Decl.), ¶208.) Botzko teaches that with multicasting, "a single selection for the entire audio section 15 can receive all SITE audio and can control multiple

switches 14 [sic], the output of which is multicast to **all SITES**." (Appx. M (Botzko), 5:63-66.)

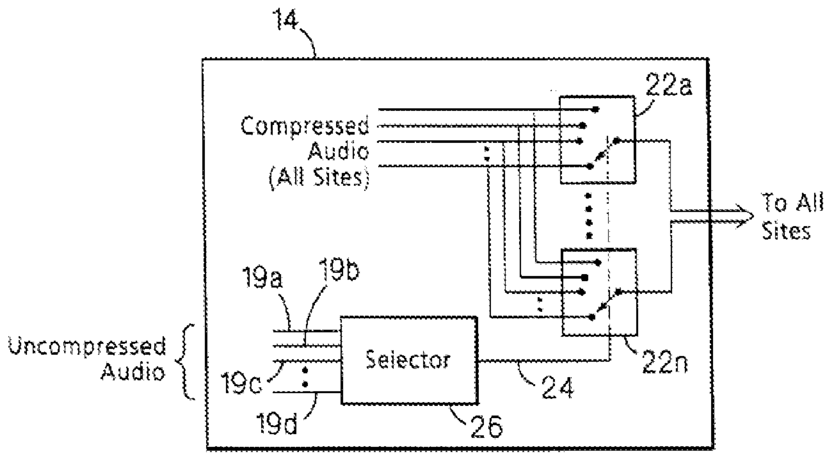


FIG. 2B

Botzko, Figure 2B

The RTP/RTCP transport circuit of Figure 2 receives multiple audio streams from the audio processors. (Appx. C (Bress Decl.), ¶¶207–08.) As described above for the two applications disclosed by Botzko, these streams are either unicast (Figure 2A) or multicast (Figure 2B) to the SITES. (Appx. M (Botzko), 6:6-14.) Therefore, the generated output audio stream from the RTP/RTCP transport circuit is a sequence of IP audio packets received from the one or more identified active speakers. (Appx. C (Bress Decl.), ¶209–10.) Thus, Botzko's RTP/RTCP transport circuit receives packets from multiple audio streams in parallel and outputs RTP/RTCP audio packets in a serial fashion. (Appx. C (Bress Decl.), ¶¶210.)

As discussed in §II.D (Claim Construction), the parties in the co-pending district court action dispute the plain and ordinary meaning (and hence the broadest **reasonable** interpretation) of the term "*multiplexed stream*." Patent Owner appears to contend that a "*multiplexed stream*" is nothing more than a sequence of audio packets. This understanding of the breadth of Patent Owner's interpretation is supported by Patent Owner's infringement contentions which appear to rely solely on transmission over TCP or UDP to show "*multiplexing said packets of audio data … into a multiplexed stream*":

1[e] (5) multiplexing said packets of audio data received from each client on said active speakers list into a multiplexed stream:	Webex practices "multiplexing said packets of audio data received from each client on said active speakers list into a multiplexed stream" for at least the reasons described *supra* at 1[a]-[d]. Webex multiplexes audio packets received from attendees into a multiplexed stream. *See* Cisco, *What's New in Cisco Webex Analytics and Troubleshooting*, available at https://help.webex.com/en-us/n45okcs/What-s-New-in-Cisco-Webex-Analytics-and-Troubleshooting (indicating audio transport configuration changes which affect the multiplexing scheme for UDP vs. TCP data streams).

(Appx. Q (Patent Owner Infringement Contentions), p. 12)

A POSITA would understand that the output stream generated by the RTP/RTCP transport circuit is a "*multiplexed stream*" under Patent Owner's interpretation of the term. (Appx. C (Bress Decl.), ¶210.) The RTP/RTCP transport circuit receives multiple output streams from switches 22 and outputs packets from each of the active speaker sources in a single IP packet stream (i.e., one packet after the other). (Appx. C (Bress Decl.), ¶210.) The RTP/RTCP transport stream in the combined system is a stream of IP packets comprised of audio packets from each client that is an active speaker. (Appx. C (Bress Decl.), ¶¶209–10.) The

155853739.8
155853739.12

output audio stream therefore meets Patent Owner's broad interpretation of "*multiplexed stream*." (Appx. C (Bress Decl.), ¶210.)

In both applications, illustrated in Figures 2A and 2B, the compressed audio selected by the switches includes "*packets of audio data received from each client on said active speakers list*." (Appx. C (Bress Decl.), ¶¶209–10.) As discussed in Section V.E.2.c ([1.2]), Botzko's selector 26 "*determin[es] which of the plurality of clients is an active speaker*." (*See, e.g.,* Appx. M (Botzko), 7:63-66 ("selected speakers will also depend upon, and typically be selected from SITES having, **a certain minimum threshold level of speech**"), Appx. M (Botzko), 5:4-6 ("an acoustic energy detector, or other device could be used as equivalent alternatives for the selector 26").) Selector 26 "produces the corresponding control signal on line 24" to select the audio for the active speakers. (Appx. M (Botzko), 4:62-66.) Each switch in Figures 2A and 2B selectively "couples one of the plurality of compressed audio signals" to the output "in accordance with and based upon a control signal on line 24." (*See, e.g.,* Appx. M (Botzko), 4:57-62; Appx. C (Bress Decl.), ¶209.) That is, the audio output from the switches is data from the active speakers. (Appx. C (Bress Decl.), ¶209.)

Accordingly, the combination of Botzko and Kumar discloses or at least suggests "*(5) multiplexing said packets of audio data received from each client on*

said active speakers list into a multiplexed stream" [1.5] under Patent Owner's broad interpretation.[7] (Appx. C (Bress Decl.), ¶¶204–10.)

f. The combination of Botzko and Kumar discloses "(6) sending said multiplexed stream to each of said first subset of the plurality of clients" [1.6].

The combination of Botzko and Kumar discloses "*(6) sending said multiplexed stream to each of said first subset of the plurality of clients*" [1.6]. (Appx. C (Bress Decl.), ¶¶212–14.)

As shown in Botzko's Figures 2 and 2B, the output of audio processor 14c (multiple audio streams of the active speakers) is sent as compressed audio to its RTP/RTCP transport circuit in bridge 12. (*See* Appx. M (Botzko), Figure 2; Appx. C (Bress Decl.), ¶213.) In Botzko, each one of the SITES "transmits and receives time compressed audio packets, here, for example, through an RTP/RTCP transport." (Appx. M (Botzko), 4:35-36.) Botzko teaches that in its multicasting application, "a single selection for the entire audio section 15 can receive all SITE audio and can control multiple switches 14 [sic], the output of which is multicast to

[7] Requestor demonstrates in Section V.F that the combination of Botzko, Kumar, and either of the RTP multiplexing references (Hoshi or Rosenberg) discloses this limitation under Requestor's interpretation of the plain and ordinary meaning.

all SITES." (Appx. M (Botzko), 5:63-66.) Thus, each SITE receives the same multiplexed output." (Appx. M (Botzko), 5:65-67.) The combination of Botzko and Kumar therefore discloses "*(6) sending said multiplexed stream to each of said first subset of the plurality of clients*" [1.6]. (Appx. C (Bress Decl.), ¶213.)

To the extent Patent Owner alleges that this claim limitation covers different unicast streams sent from a conferencing platform, Botzko discloses the limitation. In Botzko's multiple speaker/unicast application, "each end-point SITE unicasts its audio stream(s) to the conferencing bridge 12" which "selects, using selector 26, one or more streams at switches 22a, . . . , 22n, to unicast back to each end-point SITE. (See FIG. 2A)." (Appx. M (Botzko), 6:6-10; Figures 2, 2A.) In this application, an audio processor 14 associated with a SITE receives compressed audio from SITES other than the SITE it is coupled to. (*See* Appx. M (Botzko), Figure 2; Appx. C (Bress Decl.), ¶214.) This application does not include a SITE's own audio in the audio streams switched and multiplexed. Thus, Botzko also discloses sending different multiplexed streams to local mixing clients.

g. The combination of Botzko and Kumar discloses "(7) mixing said packets of audio data received from each client on said active speakers list into one combined packet" [1.7].

The combination of Botzko and Kumar discloses "*(7) mixing said packets of audio data received from each client on said active speakers list into one combined packet*" [1.7]. (Appx. C (Bress Decl.), ¶¶216–23.)

155853739.8
155853739.12

Botzko explains that "[t]here are two classic types of audio processors: an audio switch and an audio mixer." (Appx. M (Botzko), 1:13-14.) "An audio mixer operates with non-time compressed, that is, uncompressed, audio." (Appx. M (Botzko), 1:56-57.) For a SITE in a conference that cannot receive multiple streams (i.e., a non-mixing client), "the audio mixer combines the audio from selected other sites and re-encodes (that is, time compresses) the combined audio so that it can output time compressed, mixed audio to a receiving site." (Appx. M (Botzko), 1:56-61.)

Botzko's Figure 3 (reproduced below with annotations) illustrates a mixing audio processor 14'c for non-mixing SITES. As discussed in Section V.E.2.c ([1.2]), selector 26' "*determin[es] which of the plurality of clients is an active speaker*." Selector 26' "also determines whether more than one person is speaking at those sites at the same time." (Appx. M (Botzko), 7:20-23.) If only one person is speaking, mixing is not required, and therefore the mixer and encoder are not enabled. (*See* Appx. M (Botzko), 7:23-28, 7:33-39; Appx. C (Bress Decl.), ¶218.) "If more than one person is speaking at the same time, (a double-talk, triple-talk, etc., condition) a logic '1' signal is fed to line 36" and "mixer 28 and encoder 29 are enabled." (Appx. M (Botzko), 7:23-30.)

155853739.8
155853739.12

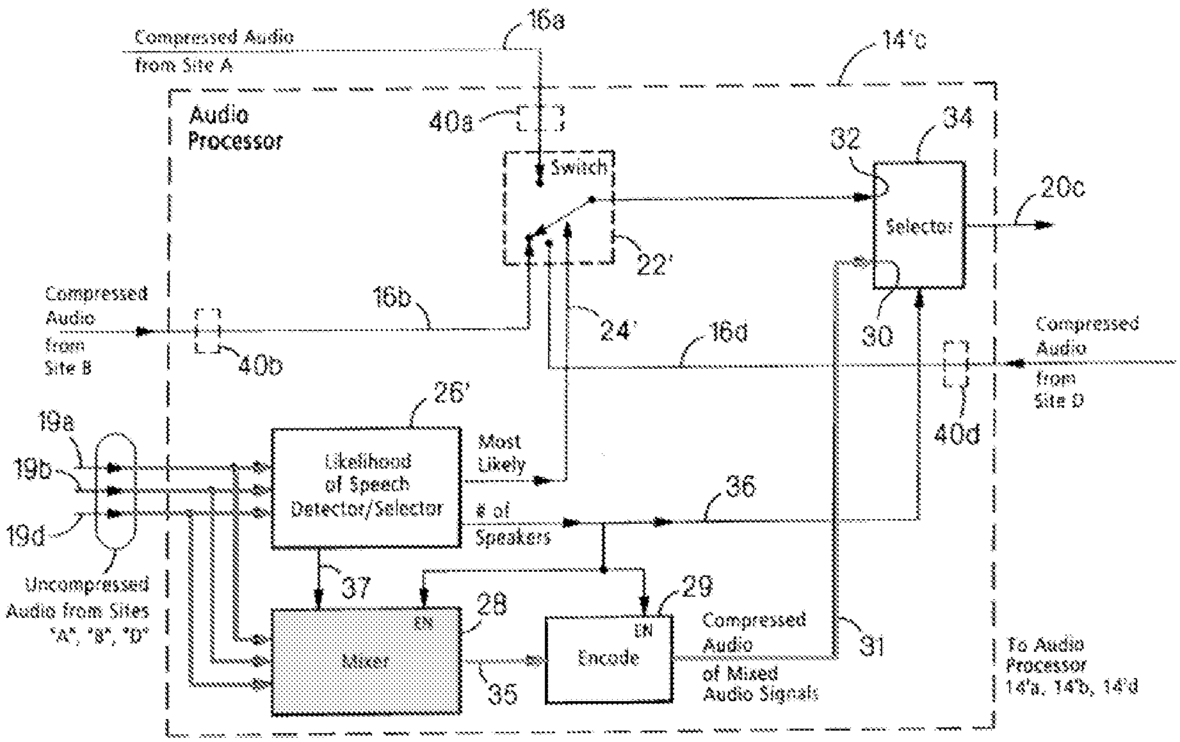


FIG. 3

Botzko, Annotated Figure 3

"When two or more speakers are determined to exist by selector 26' the detector provides to the mixer 28, over lines 37, signal information identifying on which of the input lines the speakers can be found." (Appx. M (Botzko), 7:54-57.) Mixer 28 then mixes "depending upon its configuration, 2, 3, or more inputs to produce its mixed output over line 35." (Appx. M (Botzko), 7:57-59.) The mixed output is then "encoded, that is, time compressed, in encoder 29." (Appx. M (Botzko), 7:40-45.)

Botzko illustrates an alternative embodiment having a centralized selector and set of mixers in Figure 3A (reproduced below). In this embodiment, up to three

speakers can be identified and mixed together. (Appx. M (Botzko), 8:28-30.) The selector "controls each of the mixers over lines 37 and each mixer, when enabled, produces a mixed output to an encoder 29a, 29b, 29c, 29d, respectively, which generate appropriate output signals." (Appx. M (Botzko), 8:38-42.) "The output of the encoders is directed to a cross-point switch 100." (Appx. M (Botzko), 8:42-43.)

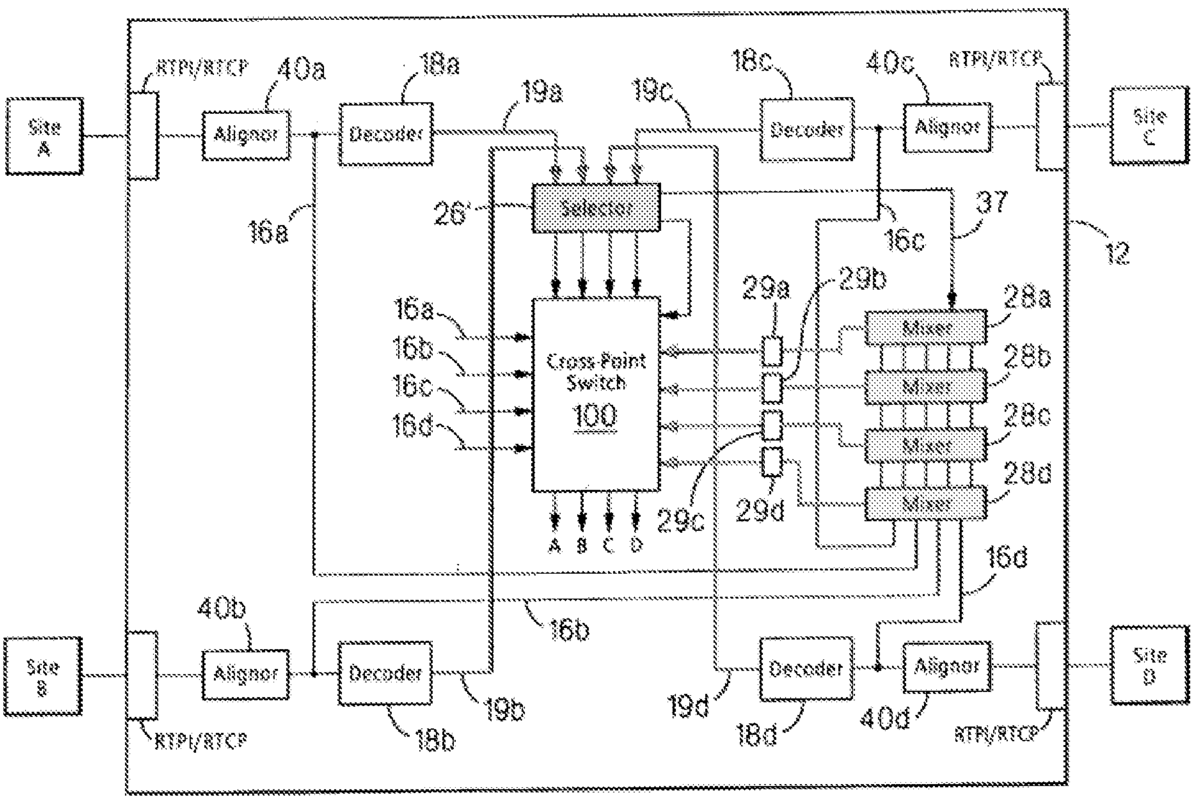


FIG. 3A

Encoders 29 "produce a corresponding compressed composite audio signal." (Appx. M (Botzko), 7:9-11.) The "mixed compressed composite audio is transmitted, through selector 34, to the end-point SITE." (Appx. M (Botzko), 7:46-47.) Botzko's audio processors 14, including audio processor 14'c, are "coupled to

a corresponding one of the sites" (e.g., SITE 'A', SITE 'B', SITE 'C', and SITE 'D') "through RTP/RTCP transport circuits." (Appx. M (Botzko), 4:2-7.) A POSITA would understand, based on Botzko's teachings, that the outputs of the cross-point switch 100 are also coupled to the SITES through RTP/RTCP transport circuits. (Appx. C (Bress Decl.), ¶221.)

Each one of the SITES "receives time compressed audio packets, here, for example, through an RTP/RTCP transport." (Appx. M (Botzko), 4:35-36.) Each mixed packet transmitted from the RTP/RTCP transport of the bridge contains all or a portion of the mixed audio and therefore is a "*combined packet*." (Appx. C (Bress Decl.), ¶223.) In conferences having local mixing and non-mixing SITES, all of the non-mixing SITES will receive the same "combined packets" with audio mixed from all active speakers when the active speakers are local mixing clients. (Appx. C (Bress Decl.), ¶222.)

The combination of Botzko and Kumar therefore discloses "*(7) mixing said packets of audio data received from each client on said active speakers list into one combined packet*" [1.7]. (Appx. C (Bress Decl.), ¶¶216–23.)

h. The combination of Botzko and Kumar discloses "(8) sending said combined packet to each of said second subset of the plurality of clients" [1.8].

The combination of Botzko and Kumar discloses "*(8) sending said combined packet to each of said second subset of the plurality of clients*" [1.8]. (Appx. C (Bress Decl.), ¶¶225–27.)

As discussed in Section V.E.2.h ([1.8]), Botzko's audio processors 14, including audio processor 14'c, are "coupled to a corresponding one of the sites" (e.g., SITE 'A', SITE 'B', SITE 'C', and SITE 'D') "through RTP/RTCP transport circuits." (Appx. M (Botzko), 4:2-7.) A POSITA would also understand, based on Botzko's teachings, that the outputs of the cross-point switch 100 are also coupled to the SITES through RTP/RTCP transport circuits. (Appx. C (Bress Decl.), ¶226.) Each one of the SITES "receives time compressed audio packets, here, for example, through an RTP/RTCP transport." (Appx. M (Botzko), 4:35-36.) The RTP/RTCP transport is part of Botzko's disclosed audio bridge and is the component responsible for sending the mixed (combined) audio packets to the non-mixing SITES (clients).

Botzko describes an example of a conference with three speakers. In this example, four encoders and four mixers "are needed (no matter how many sites are connected)." (Appx. M (Botzko), 8:18-20.) This arrangement "allows the loudest speaker to hear the two next loudest speakers"; "the second loudest speaker to hear

the loudest speaker and the third loudest speaker"; and "the third loudest speaker hears a mix of the two loudest speakers." (Appx. M (Botzko), 8:20-24.) "Everyone else hears a mix of the three loudest speakers." (Appx. M (Botzko), 8:24-25.) That is, multiple SITES in the conference receive the same "combined packets." And, in conferences having local mixing and non-mixing SITES, all of the non-mixing SITES will receive the same "combined packets" with audio mixed from all active speakers when the active speakers are local mixing clients. (Appx. C (Bress Decl.), ¶227.)

Botzko therefore teaches or suggests "*(8) sending said combined packet to each of said second subset of the plurality of clients*" [1.8]. (Appx. C (Bress Decl.), ¶¶225–27.)

i. The combination of Botzko and Kumar discloses "whereby said plurality of clients can simultaneously participate in a single audio conference application" [1.9].

The limitation "*whereby said plurality of clients can simultaneously participate in single audio conference application*" should not be afforded any patentable weight because it merely expresses the intended result of the process steps of claim. *See Hoffer v. Microsoft*, 405 F.3d 1326, 1329 (Fed. Cir. 2005) (noting that a "whereby clause in a method claim is not given weight when it simply expresses the intended result of a process step positively recited."). Regardless, the combination of Botzko and Kumar discloses this claim element.

Botzko is directed to audio processors which are "responsible for receiving audio from various sites connected to the conference system and for distributing the audio to the various sites." (Appx. M (Botzko), 1:4-12.) Botzko's bridge 12 "operates to selectively forward and/or mix, the audio from the various SITES so that each SITE can participate in a conference." (Appx. M (Botzko), 4:10-12.) Thus, Botzko discloses that "*said plurality of clients can simultaneously participate in a single audio conference application.*" (Appx. C (Bress Decl.), ¶¶229–30.)

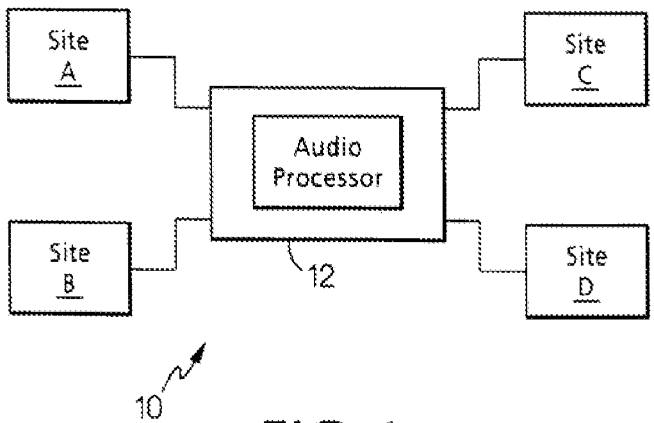
3. The combination of Botzko and Kumar renders independent claim 6 obvious.

a. The combination of Botzko and Kumar discloses the preamble [6P].

To the extent the preamble is limiting, Botzko discloses a "*system for providing audio conferencing for a plurality of clients*" [6P]. (Appx. C (Bress Decl.), ¶¶186–89.)

Botzko is directed to audio processors which are "responsible for receiving audio from various sites connected to the conference system and for distributing the audio to the various sites." (Appx. M (Botzko), 1:4-12.) Botzko's "audio conferencing system", illustrated in Figure 1 below, includes a plurality of SITES (e.g., SITES A, B, C, and D) "connected together through a server, or bridge 12." (Appx. M (Botzko), 3:65-4:2.) Bridge 12 "operates to selectively forward, and/or

mix, the audio from the various SITES so that each SITE can participate in a conference." (Appx. M (Botzko), 4:10-12.) Botzko's SITES are the recited "*plurality of clients*."



Botzko, Figure 1

b. The combination of Botzko and Kumar discloses "a receiver capable of receiving an audio packet from each of the plurality of clients" [6.1].

Botzko discloses "*a receiver capable of receiving an audio packet from each of the plurality of client*" [6.1]. (Appx. C (Bress Decl.), ¶¶190–91.) For the reasons explained above with respect to claim 1, the combination of Botzko and Kumar discloses "*receiving an audio packet from each of the plurality of clients.*" (*See supra*, Section V.E.2.b.) In Botzko, "each one of the SITES 'A' - 'D', **transmits** and receives **time compressed audio packets**, here, for example, through an RTP/RTCP transport." (Appx. M (Botzko), 4:34-36; Figure 2 (below).) As shown in Figure 2 (below), the audio processors of the bridge are coupled "to a

corresponding one of the sites" "through RTP/RTCP transport circuits." (Appx. M (Botzko), 4:2-7.) Botzko's RTP/RTCP transport circuits are the recited "*receiver*." (Appx. C (Bress Decl.), ¶191.)

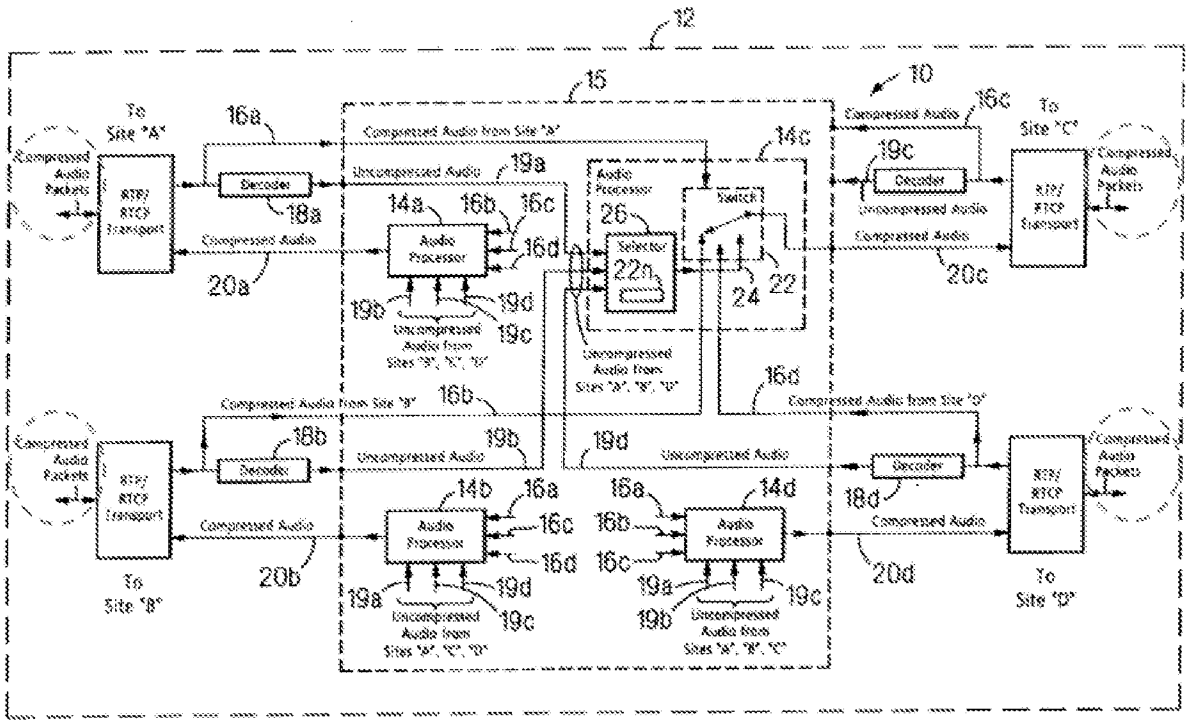


FIG. 2

Botzko, Annotated Figure 2

c. The combination of Botzko and Kumar teaches or at least suggests "means for maintaining a list of each of the plurality of clients that is an active speaker" [6.2].

The combination of Botzko and Kumar teaches or at least suggests a "*means for maintaining a list of each of the plurality of clients that is an active speaker*" [6.2]. (Appx. C (Bress Decl.), ¶¶192–99.) As discussed in the Claim Construction section, the parties in the co-pending district court action agree that the function of

155853739.8
155853739.12

this limitation is "maintaining a list of each of the plurality of the clients that is an active speaker" and the structure is "control logic, like that of control flow 300 in Figure 3, executed by the computer system 400 in Figure 4 as well as equivalents thereof." (*See supra* Section II.D.3.)

As discussed in Section V.E.2.c ([1.2]), Botzko discloses "*determining which of the plurality of clients is an active speaker*" and teaches or at least suggests "*forming an active speakers list*." Because Botzko's bridge utilizes the identification of the active speakers to control its mixers, multiplexers, and the cross-point switch (Figure 3A), a POSITA would have understood that the bridge "*maintains*" the active speaker list for the duration of a conference. (Appx. C (Bress Decl.), ¶199.) Because Botzko teaches that its "invention operates primarily in software," a POSITA would have been motivated to maintain the list of active speakers using control logic. (*See* Appx. M (Botzko), 4:14-18, 8:54-57.)

The combination of Botzko and Kumar therefore teaches or at least suggests "*means for maintaining a list of each of the plurality of clients that is an active speaker*" [6.3]. (Appx. C (Bress Decl.), ¶¶192–99.)

d. The combination of Botzko and Kumar teaches or at least suggests "means for storing information indicative of whether each of the plurality of clients has the capability to mix multiple audio streams" [6.3].

The combination of Botzko and Kumar discloses or at least suggests "*means for storing information indicative of whether each of the plurality of clients has the*

capability to mix multiple audio streams" [6.3]. (Appx. C (Bress Decl.), ¶¶200–03.) As discussed in the Section II.D.3, the parties in the co-pending district court action agree that this limitation requires the function of "storing information indicative of whether each of the plurality of clients has the capability to mix multiple audio streams" to be performed by the structure of a "main memory 408 and processor 404 in Figure 4 as well as equivalents thereof."

As discussed in Section V.E.2.d ([1.3] and [1.4]), in the combination of Botzko and Kumar, Botzko's bridge receives information regarding the capabilities of the SITES through a capabilities exchange, such as taught by Kumar. Specifically, Kumar discloses that an end-point terminal provides an indication of its audio mixing capability—i.e., how many simultaneous audio streams it is capable of decoding. (Appx. P (H.323 11/96, 13.) The audio mixing capability is "*information indicative of whether each of the plurality of clients has the capability to mix multiple audio streams*" as discussed in Section V.E.2.d.

Botzko further references the "bridge configuration." (Appx. M (Botzko), 7:59-60 ("The level of mixing will depend upon the bridge configuration, including, in particular, the number of connected SITES and the desirability of hearing more than two or three speakers at the same time.").) The mixing capabilities of a SITE determine the type of audio processor associated with the SITE which is part of the bridge configuration. (Appx. C (Bress Decl.), ¶202.)

Although Botzko does not explicitly mention that the bridge configuration is "stored", it would have been obvious to a POSITA that configuration information would be stored to set-up and operate the bridge. (Appx. C (Bress Decl.), ¶203.)

Botzko does not explicitly mention that its bridge includes memory and a processor. However, because Botzko teaches that its "invention operates primarily in software," a POSITA would have understood the bridge uses a processor and memory. (*See* Appx. M (Botzko), 4:14-18, 8:54-57.) Moreover, Kumar discloses a device including memory and a processor for running software. (*See* Appx. N (Kumar), 5:21-24.) Accordingly, a POSITA would have been motivated, based on the suggestions of Kumar and Botzko, to store the capabilities information for a SITE in a memory at the bridge using a processor. (Appx. C (Bress Decl.), ¶203.)

The combination of Botzko and Kumar discloses or at least suggests "*means for storing information indicative of whether each of the plurality of clients has the capability to mix multiple audio streams*" [6.3]. (Appx. C (Bress Decl.), ¶¶200–03.)

e. The combination of Botzko and Kumar teaches or at least suggests "a multiplexor capable of multiplexing said packets of audio data received from each client on said list of active speakers into a multiplexed stream" [6.4].

The combination of Botzko and Kumar teaches or at least suggests "*a multiplexor capable of multiplexing said packets of audio data received from each client on said list of active speakers into a multiplexed stream*" [6.4] under Patent

Owner's interpretation of "*multiplexed stream*". (Appx. C (Bress Decl.), ¶¶204–11.) As discussed in Section V.E.2.e ([1.5]), Botzko discloses "*multiplexing said packets of audio data received from each client on said active speakers list into a multiplexed stream*" under Patent Owner's broad interpretation of the term in the co-pending district court limitation.

A POSITA would understand that the RTP/RTCP transport circuit in Botzko's bridge includes a "*multiplexor capable of*" performing "*multiplexing.*" The term "multiplexor" is not defined in the specification and therefore should be interpreted as simply a component that perform "*multiplexing.*" In Botzko, as discussed in Section V.E.2.e, the RTP/RTCP transport circuit receives packets from multiple audio streams of the active speakers in parallel and outputs RTP/RTCP audio packets in a serial fashion to a SITE. (Appx. C (Bress Decl.), ¶210.) A POSITA would therefore understand that Botzko's RTP/RTCP transport circuit includes a component that is "*capable of multiplexing said packets of audio data received from each client on said list of active speakers into a multiplexed stream.*" (Appx. C (Bress Decl.), ¶211.)

Accordingly, combination of Botzko and Kumar teaches or at least suggests "*a multiplexor capable of multiplexing said packets of audio data received from each client on said list of active speakers into a multiplexed stream*" [6.4] under Patent Owner's broad interpretation. (Appx. C (Bress Decl.), ¶¶204–11.)

f. The combination of Botzko and Kumar discloses "a mixer capable of mixing said packets of audio data received from each client on said list of active speakers into one combined packet" [6.5].

The combination of Botzko and Kumar discloses "*a mixer capable of mixing said packets of audio data received from each client on said list of active speakers into one combined packet* [6.5]. (Appx. C (Bress Decl.), ¶¶216–24.) As discussed in Section V.E.2.g ([1.7]), the combination of Botzko and Kumar discloses "*mixing said packets of audio data received from each client on said list of active speakers into one combined packet*."

Botzko explains that "[t]here are two classic types of audio processors: an audio switch and an audio mixer." (Appx. M (Botzko), 1:13-14.) "An audio mixer operates with non-time compressed, that is, uncompressed, audio." (Appx. M (Botzko), 1:56-57.) For a SITE in a conference that cannot receive multiple streams (i.e., a non-mixing client), "the audio mixer combines the audio from selected other sites and re-encodes (that is, time compresses) the combined audio so that it can output time compressed, mixed audio to a receiving site." (Appx. M (Botzko), 1:56-61.) Botzko's mixing audio processor 14'c for non-mixing SITES include a mixer 28, as shown in Figure 3 below, that performs audio "mixing". (*See, e.g.,* Appx. M (Botzko), 7:57-59; *see also*, 8:29-60, Figure 3A.) In conferences having local mixing and non-mixing SITES, all of the non-mixing SITES will receive the same "combined packets" with audio mixed from all active

speakers when the active speakers are local mixing clients. (Appx. C (Bress Decl.), ¶222.)

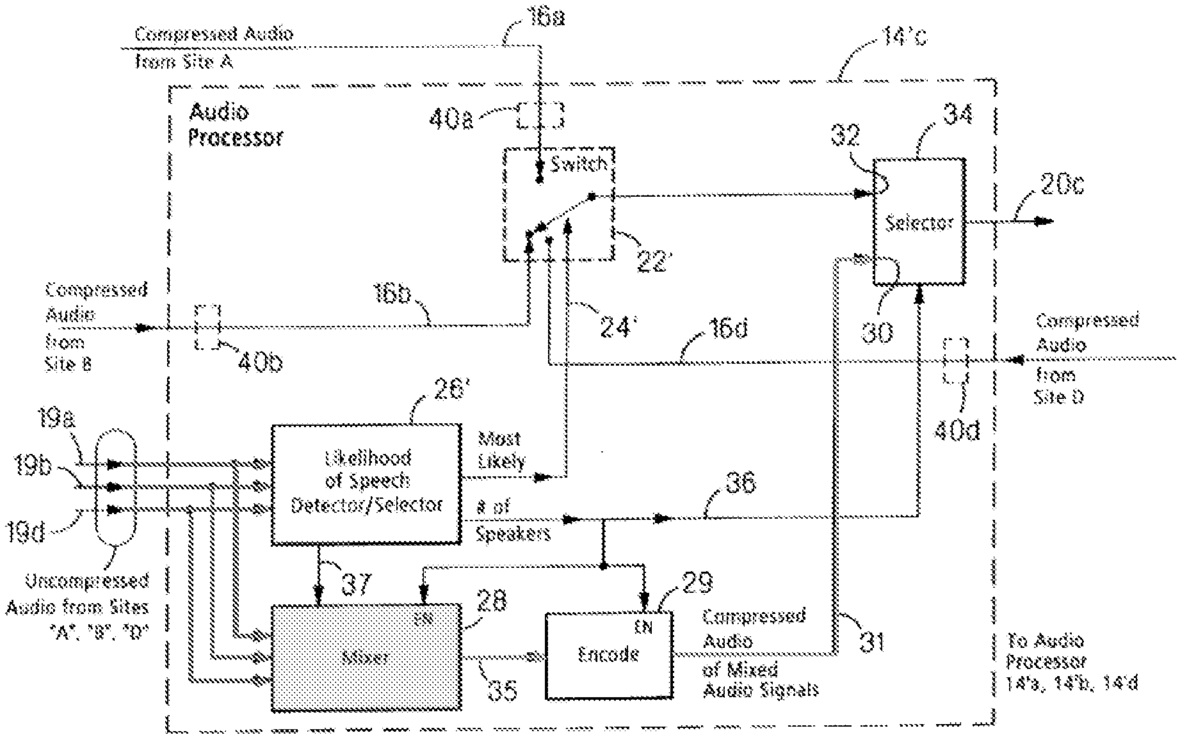


FIG. 3

Botzko, Annotated Figure 3

Accordingly, the combination of Botzko and Kumar discloses "*a mixer capable of mixing said packets of audio data received from each client on said list of active speakers into one combined packet*" [6.5]. (Appx. C (Bress Decl.), ¶¶216–44.)

g. The combination of Botzko and Kumar discloses "*a packet sender capable of sending …*" [6.6].

The combination of Botzko and Kumar discloses "*a packet sender capable of sending, based on information in said means for storing, said multiplexed*

stream to each of the plurality of clients which have the capability to mix multiple audio streams, and capable of sending said combined packet to each of the plurality of clients which do not have the capability to mix multiple streams" [6.6]. (Appx. C (Bress Decl.), ¶¶212–15, 225–28.)

As discussed in Sections V.E.2.f ([1.6]) and V.E.2.h ([1.8]), the combination of Botzko and Kumar discloses "*sending said multiplexed stream to each of said first subset of clients*" and "*sending said combined packet to each of said second subset of the plurality of clients*." In claim 1, the "*first subset of clients*" is the subset of clients that has been determined to have "*the capability to mix multiple audio streams*" and the "*second subset of the plurality of clients*" is the subset of clients that has been determined to "*not have the capability to mix multiple audio streams*." Thus, for the reasons discussed in claim 1, the combination of Botzko and Kumar discloses that the "*sending*" of the "*multiplexed stream*" and "*combined packet*" is based on determined capability of a SITE. And, as discussed in Section V.E.3.d ([6.3], this information is stored in Botzko's platform in the combination of Botzko and Kumar.

As discussed in Section V.E.2.h ([1.8]), Botzko's audio processors 14, including audio processor 14'c, are "coupled to a corresponding one of the sites" (e.g., SITE 'A', SITE 'B', SITE 'C', and SITE 'D') "through RTP/RTCP transport circuits." (Appx. M (Botzko), 4:2-7.) A POSITA would also understand, based on

Botzko's teachings, that the outputs of the cross-point switch 100 are also coupled to the SITES through RTP/RTCP transport circuits. (Appx. C (Bress Decl.), ¶212.) Each one of the SITES "receives time compressed audio packets, here, for example, through an RTP/RTCP transport." (Appx. M (Botzko), 4:35-36.) Botzko's RTP/RTCP transport circuits are therefore "*a packet sender capable of sending … said multiplexed stream*" and "*capable of sending said combined packets* …." (Appx. C (Bress Decl.), ¶¶215, 228.)

Accordingly, the combination of Botzko and Kumar discloses "*a packet sender capable of sending, based on information in said means for storing, said multiplexed stream to each of the plurality of clients which have the capability to mix multiple audio streams, and capable of sending said combined packet to each of the plurality of clients which do not have the capability to mix multiple streams*" [6.6]. (Appx. C (Bress Decl.), ¶¶212–15, 225–28.)

h. The combination of Botzko and Kumar discloses "whereby the plurality of clients can simultaneously participate in a single audio conference application" [6.7].

The limitation "*whereby said plurality of clients can simultaneously participate in a single audio conference application*" should not be afforded any patentable weight because it merely expresses the intended result of the process steps of claim. *See Hoffer v. Microsoft*, 405 F.3d 1326, 1329 (Fed. Cir. 2005) (noting that a "whereby clause in a method claim is not given weight when it

155853739.8
155853739.12

simply expresses the intended result of a process step positively recited.”). Regardless, the combination of Botzko and Kumar discloses this claim element.

Botzko is directed to audio processors which are “responsible for receiving audio from various sites connected to the conference system and for distributing the audio to the various sites.” (Appx. M (Botzko), 1:4-12.) Botzko’s bridge 12 “operates to selectively forward and/or mix, the audio from the various SITES so that each SITE can participate in a conference.” (Appx. M (Botzko), 4:10-12.) Thus, Botzko discloses that “*said plurality of clients can simultaneously participate in a single audio conference application.*” (Appx. C (Bress Decl.), ¶¶229–30.)

4. The combination of Botzko and Kumar renders dependent claims 2 and 7 obvious.

The combination of Botzko and Kumar discloses or suggests “*before sending said multiplexed stream to one of said first subset of the plurality of clients, removing from said multiplexed stream said packets of audio data received from said one of said first subset of the plurality of clients when said one of said first subset of the plurality of clients is on said active speakers list*” as recited in dependent claim 2 (Appx. C (Bress Decl.), ¶¶231–34) and “*means for removing, before said packet sender sends said multiplexed stream to one of the plurality of clients which have the capability to mix multiple audio streams, from said multiplexed stream said packets of audio data received from said one of the*

plurality of clients, when said one of the plurality of clients is on said list of active speakers" as recited in claim 7 (Appx. C (Bress Decl.), ¶231–34).

As discussed in the analysis of claim 1, Botzko describes two applications having multiple output streams. In the multicasting application, "each end-point SITE unicasts its audio stream(s) to the conferencing bridge 12" which "selects, using selector 26, one or more streams at switches 22a, . . . , 22n, and multicasts the selected streams to all SITES." (Appx. M (Botzko), 6:10-14.) The streams selected by selector 26 to stream to all SITES are the audio streams of the active speakers as described previously for claim elements [1.2] and [1.5].In this application, audio processor 14 receives compressed audio from all SITES, including the SITE it is coupled to. (*See* Appx. M (Botzko), Figure 2; Appx. C (Bress Decl.), ¶232.) Botzko teaches that in the multicasting application, "a single selection for the entire audio section 15 can receive all SITE audio and can control multiple switches 14 [sic], the output of which is multicast to all SITES." (Appx. M (Botzko), 5:63-66.) Because each SITE receives the same multiplexed stream, "each end-point SITE must automatically ignore its own transmitted stream." (Appx. M (Botzko), 5:66-67.) Thus, Botzko teaches that audio packets from one of the SITES is "*removed from said multiplexed stream*." (Appx. C (Bress C), ¶¶231–32.)

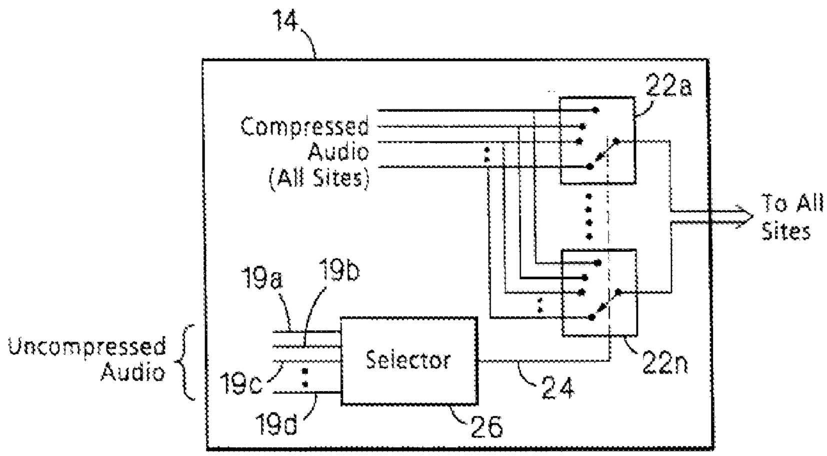


FIG. 2B

Botzko, Figure 2B

A POSITA would have been motivated to perform the actions needed to "ignore" a SITES own packets at multiple places in the network. In a first option, a POSITA would have been motivated to "ignore" a SITES' own packets by removing them from the received "*multiplexed stream*" at a client device before "*sending*" the audio packets to the local application for mixing, as suggested in Botzko. (Appx. C (Bress Decl.), ¶233.) This option takes full advantage of the benefits of multiplexing and addresses the risk of echo created by mixing in a client's own audio stream. (Appx. C (Bress Decl.), ¶233.) In a second option, a POSITA would have been motivated to remove a SITES own packets from the "*multiplexed stream*" (under Patent Owner's interpretation) at the last multicast router before the stream is sent to the SITE. (Appx. C (Bress Decl.), ¶233.) Indeed, this option was a defined part of the multicasting standard prior to the '858 patent.

(Appx. C (Bress Decl.), ¶233, *citing* Appx. AA (IETF RFC 2462). Botzko suggests this option noting that in a supported application "each end-point SITE multicasts its audio stream(s)" and the "bridge selects one or more streams, and multicasts them on a separate multicast address." (Appx. M (Botzko), 6:14-18.) In both options, a client's own packets of audio data are removed before sending the "*multiplexed stream*" to either the client device or client conferencing application. (Appx. C (Bress Decl.), ¶¶233.)

Accordingly, the combination of Botzko and Kumar discloses or suggests "*before sending said multiplexed stream to one of said first subset of the plurality of clients, removing from said multiplexed stream said packets of audio data received from said one of said first subset of the plurality of clients when said one of said first subset of the plurality of clients is on said active speakers list*" [2] and "*means for removing, before said packet sender sends said multiplexed stream to one of the plurality of clients which have the capability to mix multiple audio streams, from said multiplexed stream said packets of audio data received from said one of the plurality of clients, when said one of the plurality of clients is on said list of active speakers*" [7].

Should Patent Owner contend that the claimed removal can happen before the multiplexed stream is created, Botzko discloses the limitation as well. In the multiple speaker/unicasting application, "each end-point SITE unicasts its audio

stream(s) to the conferencing bridge 12" which "selects, using selector 26, one or more streams at switches 22a, . . . , 22n, to unicast back to each end-point SITE. (See FIG. 2A)." (Appx. M (Botzko), 6:6-10.) In this application, audio processor 14 receives compressed audio from SITES other than the SITE it is coupled to. (*See* Appx. M (Botzko), Figure 2; Appx. C (Bress Decl.), ¶234.) Thus, the audio processor in this application does not include a SITE's own audio stream in the audio streams being switched and multiplexed. As shown in Botzko's Figures 2 and 2A, the output of audio processor 14c (multiple audio streams) is sent as compressed audio to its RTP/RTCP transport circuit in bridge 12. (*See* Appx. M (Botzko), Figure 2; Appx. C (Bress Decl.), ¶234.) Thus, should these claims be interpreted as covering the scenario where audio from a SITE is not included in the generated "*multiplexed stream*" (rather than being removed from the generated "*multiplexed stream*"), Botzko renders the claims obvious.

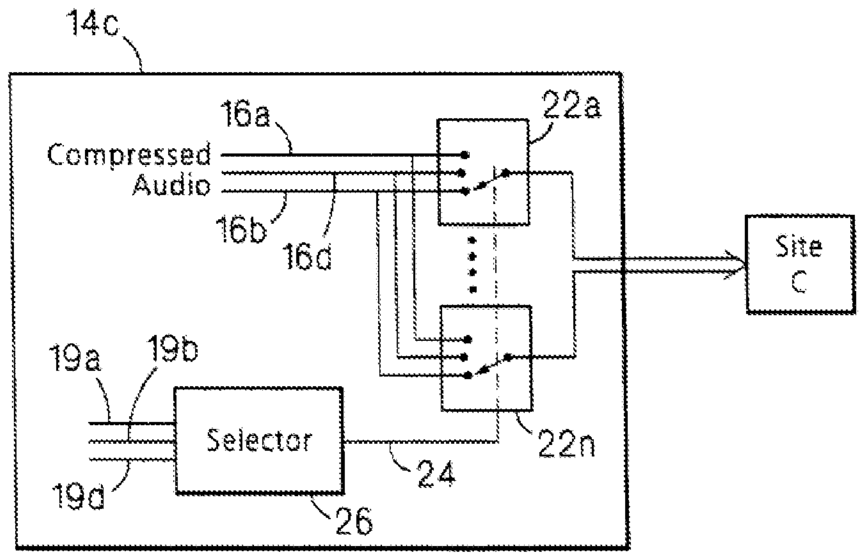


FIG. 2A

Botzko, Figure 2A

5. The combination of Botzko and Kumar renders claims 3 and 8 obvious under Patent Owner's interpretation.

Claim 3 recites "*before sending said combined packet to one of said second subset of the plurality of clients, removing from said combined packet said packets of audio data received from said one of said second subset of the plurality of clients when said one of said second subset of the plurality of clients is on said active speakers list*" and claim 8 recites "*means for removing, before said packet sender sends said combined packet to one of the plurality of clients which do not have the capability to mix multiple audio streams, from said combined packet said packets of audio data received from said one of the plurality of clients, when said one of the plurality of clients is on said list of active speakers.*" Should Patent

155853739.8
155853739.12

Owner contend that the removal step of claims 3 and 8 occurs prior to the creation of the combined packet, the combination of Botzko and Kumar discloses these limitations.

Botzko's Figure 3 (reproduced below with annotations) illustrates a mixing audio processor 14'c for non-mixing SITES. In the mixing audio processor shown in Figure 3 for SITE C, "uncompressed audio signals on lines 19a, 19b and 19d from SITES 'A', 'B', and 'D', are also fed to an audio mixer 28 to produce an uncompressed composite audio signal over a line 35." (Appx. M (Botzko), 7:5-8.) And "[i]f more than one person is speaking at the same time, (a double-talk, triple-talk, etc., condition) a logic '1' signal is fed to line 36" and "mixer 28 and encoder 29 are enabled." (Appx. M (Botzko), 7:23-30.) The streams selected to be mixed are the audio streams of the active speakers as described previously for claim elements [1.2] and [1.5]. The mixed, uncompressed audio signal, is then "fed to a time compression encoder 29 to produce a corresponding compressed composite audio signal over a line 31." (Appx. M (Botzko), 7:8-11.) Because the uncompressed audio for SITE C is not provided to the mixer, the resulting mixed audio does not have SITE C's audio—it is not included in the combined packet. (Appx. C (Bress Decl.), ¶236.)

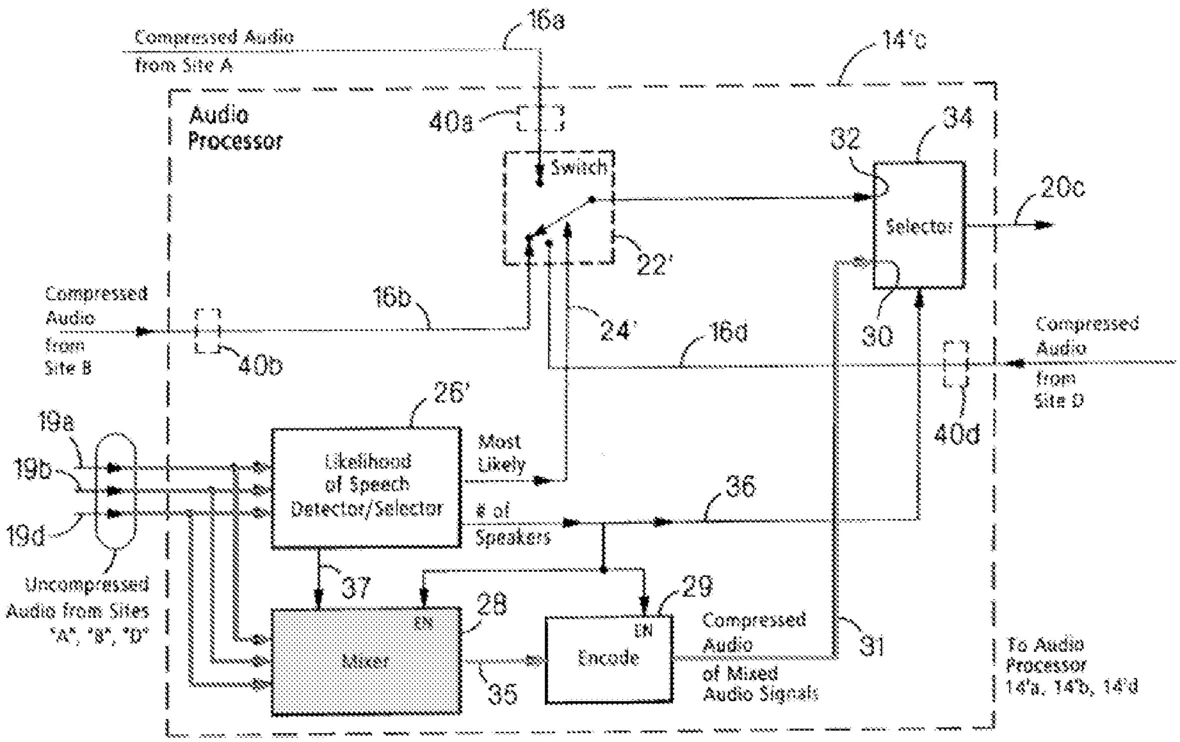


FIG. 3

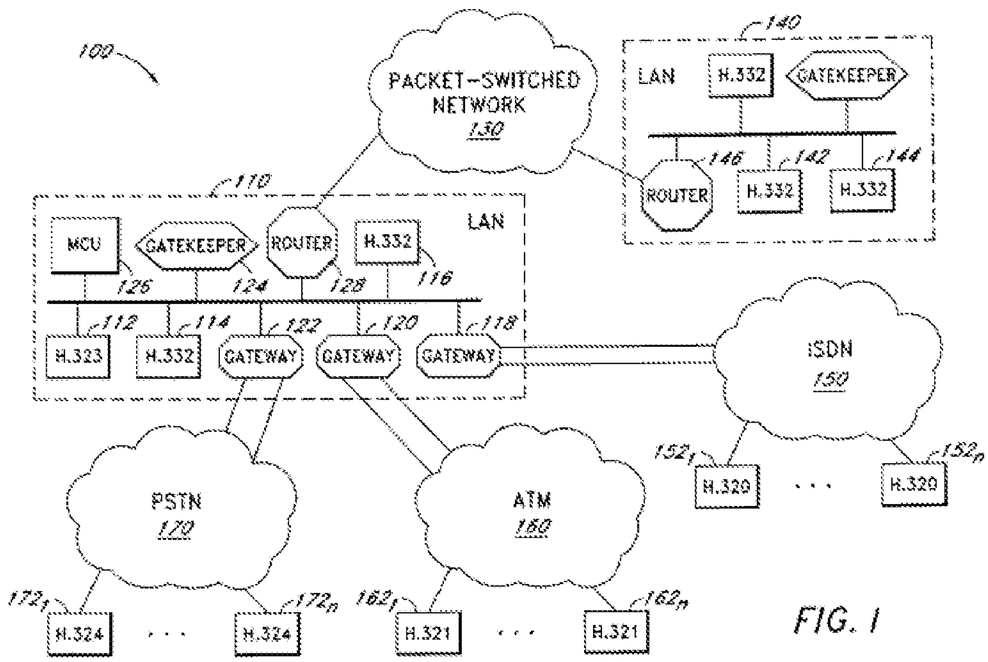
6. The combination of Botzko and Kumar renders dependent claims 5 and 10 obvious.

The combination of Botzko and Kumar discloses that "*at least one of said second subset of the plurality of clients is using a telephone and the H.323 protocol*" as recited in dependent claim 5 and that "*at least one of the plurality of clients, which does not have the capability to mix multiple audio streams, is using a telephone and the H.323 protocol*" as recited in dependent claim 10. (Appx. C (Bress Decl.), ¶¶237–40.)

The detailed description of the '858 patent provides only brief mentions of H.323. (*E.g.*, Appx. A ('858 patent), 4:7–11 ("The Switch 114 enables the service provider's MCU 116 to receive audio packets from both PC-based clients 102

using, for example, the SIP protocol, as well as receive H.323 protocol packets from the telephone-based clients 108 who connect via gateway 112."), 5:29–31 ("Whether a particular party is a mixing client (e.g., a PC-based client 102 using SIP) or not (e.g., a telephone client 108 using H.323)…").) Telephone 108 is coupled to gateway 112 through a circuit-switched network. (Appx. A ('858 patent), 3:66–41.) Kumar discloses telephone clients using H.323 in a manner consistent with the disclosure of the '858 patent.

Kumar's system, illustrated in Figure 1 below, includes a plurality of terminals including H.324 terminals 172_1-172_p coupled to a gateway 122 via the PSTN. (*See* Appx. N (Kumar), 3:25-38.) Botzko's conferencing system supports normal end-points "which can receive one audio stream to play out of their loudspeakers." (Appx. M (Botzko), 5:23-25.) A POSITA would have understood that one or more terminals coupled to the PSTN in the combination of Botzko and Kumar are normal end-points (i.e., non-mixing endpoints.)



Kumar, Figure 1

Because Kumar's H.324 terminals are coupled to the PSTN, and coupled to a. H.323 gateway to connect to H.323 terminals and other H.323 endpoints, a POSITA would understand that these H.324 (PSTN) terminals used the H.323 protocol (at least in the gateway) in the same manner as disclosed in the '858 patent: "a telephone client 108 using H.323." (Appx. C (Bress Decl.), ¶240.) Therefore, the combination of Botzko and Kumar teaches or at least suggest that "*at least one of said second subset of the plurality of clients is using a telephone and the H.323 protocol*" [5] and "*at least one of the plurality of clients, which does not have the capability to mix multiple audio streams, is using a telephone and the H.323 protocol*" [10]. (Appx. C (Bress Decl.), ¶¶237–40.)

F. GROUND XI-XII: The combination of Botzko, Kumar, and one of the RTP multiplexing references [Hoshi or Rosenberg] renders claims 1-10 obvious.

As discussed in Section V.E, the combination of Botzko and Kumar discloses the preambles and limitations [1.1]–[1.4], [1.7]–[1.9], [6.1]–[6.3] and [6.5]-[6.7] of claims 1 and 6. The combination further discloses the "*multiplexing*"/"*multiplexor*" limitations [1.5]/[6.4] under Patent Owner's interpretation of the plain and ordinary meaning of the term "*multiplexed stream*." However, the combination of Botzko and Kumar does not explicitly disclose these limitations under Requester's interpretation of the plain and ordinary meaning of "*multiplexed stream*" in the co-pending district court action: "a data structure containing a continuous sequence of interleaved packets of audio data from each client on the active speakers list." (*See* Appx. D (Cisco Claim Construction Brief), p. 5.) As discussed in Section V.E.2.e, the combination of Botzko and Kumar teaches generation of "a continuous sequence of interleaved packets of audio data from each client on the active speakers list." However, the combination is silent on how the transmission is achieved. (Appx. C (Bress Decl.), ¶241.) Specifically, neither Botzko and Kumar explicitly discloses that the "continuous sequence of interleaved packets of audio data from each client on the active speakers list" is contained in a "data structure."

The rudimentary concept of transmitting multiplexed data in a data structure is disclosed in both Hoshi and Rosenberg as detailed below. The additional combinations presented in this ground render independent claims 1 and 6 and dependent claims 4-5 and 9-10 unpatentable as set forth below:

- GROUND XI: The combination of Botzko, Kumar and Hoshi renders claims 1-3, 5-6, and 10 unpatentable.
- GROUND XII: The combination of Botzko, Kumar and Rosenberg renders claims 1-10 unpatentable.

1. Overview of the Combinations

a. Overview of Hoshi

"Proposal of a Method of for [sic] Voice Stream Multiplexing for IP Telephony Systems" by Hoshi, et al (Appx. I ("Hoshi") was published in the IEEE Conference Proceedings for the 1999 Internet Workshop, IWS 99, held on February 18–20, 1999. (*See* Appx. R (Proceedings Front Cover); *see also* Appx. S (LOC MARC record) (indicating October 10, 1999 receipt date for proceedings in MARC entry 008).) Hoshi was also cited on the face of U.S. Patent No. 7,158,491 (Appx. T), which claims priority to U.S. Provisional Application No. 60/163,583, filed November 5, 1999, and U.S. Patent No. 8,189,592 (Appx. U), which claims priority to U.S. Provisional Application No. 60/209,551, filed June 6, 2000. Hoshi is prior art under at least pre-AIA 35 U.S.C. §§ 102(a) and (b).

Hoshi recognized that "for packet transfer over the IP network, it is necessary to add packet headers, including IP, UDP, and RTP headers and these cause an overhead resulting in inefficient bandwidth usage." (Appx. I (Hoshi), p. 182.) And moreover, "because there will be large numbers of short voice packets flowing into the IP network, the load on the Internet will increase." (Appx. I (Hoshi), p. 182.) Hoshi therefore proposed a technique for "voice stream multiplexing between IP-GWs to solve these problems." (Appx. I (Hoshi), p. 182.)

Hoshi's voice stream multiplexing "concatenate[s] RTP voice packets from the streams to the same destination IP-GWs onto a single UDP packet at a multiplexing interval period (Appx. I (Hoshi), p. 183.) Figure 3 of Hoshi (reproduced below) illustrates a "system configuration of an IP telephony system with voice stream multiplexing." (Appx. I (Hoshi), p. 184.) In this configuration, in addition to single-stream voice channels, multiplexed channels are provided between network components (e.g., IP-GWs). (Appx. I (Hoshi), p. 184.)

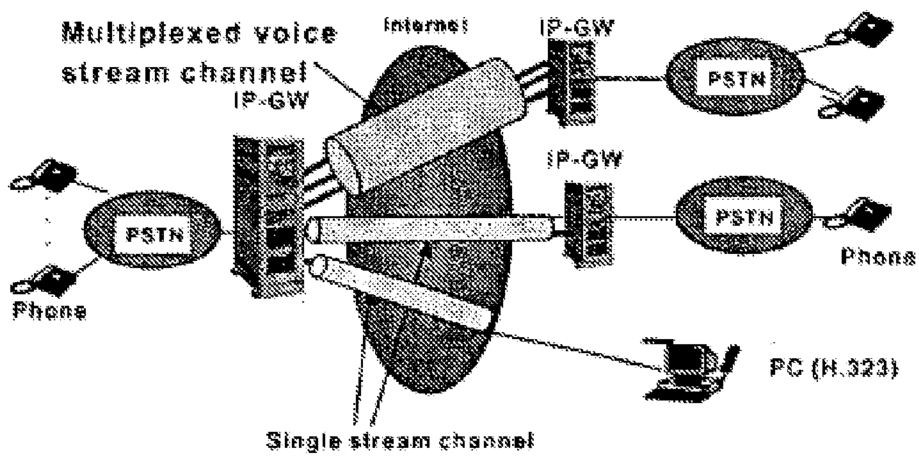
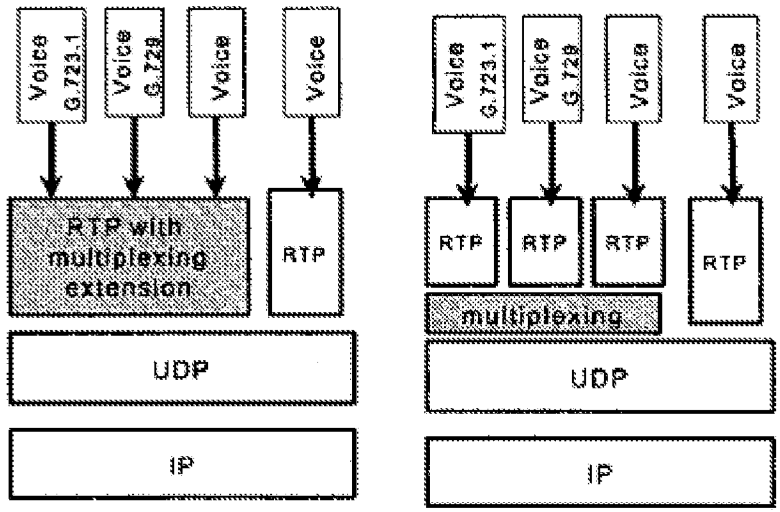


Fig. 3 IP telephony with voice stream multiplexing

Hoshi presents two techniques for its voice stream multiplexing. (Appx. I (Hoshi), p. 184, Figure 4 (below).) In the first technique, "multiplexing is done in the RTP layer so that voice frames from each voice stream are multiplexed and encapsulated into an RTP packet." (Appx. I (Hoshi), p. 184.) In the second technique, "multiplexing RTP voice packets into a UDP frame, the multiplexing layer is between the RTP and UDP layers." (Appx. I (Hoshi), p. 184.)

Fig. 4 Multiplexing layer



(a) Multiplexing in RTP (b) Multiplexing RTP packets

In Hoshi, a multiplexing interval "defines the period when RTP packets are multiplexed" to create a "multiplexed voice packet." (Appx. I (Hoshi), p. 185.) The multiplexing interval timing is a settable parameter. (Appx. I (Hoshi), p. 185.) If the timing is set to be large, "the number of RTP packets in a multiplexed packet becomes large" which "reduces both the header overhead and the number of packets, but increases the multiplexing delay." (Appx. I (Hoshi), p. 185.) If the timing is set to be small, "the number of RTP packets in a multiplexed packet becomes small, which results in smaller reductions of both header overhead and packet number." (Appx. I (Hoshi), p. 185.) After the interval timing has expired, a "multiplexed voice packet is composed by concatenating RTP-encapsulated voice packets and IP and UDP headers" as illustrated in Figure 5 below. (Appx. I (Hoshi), p. 184–185.)

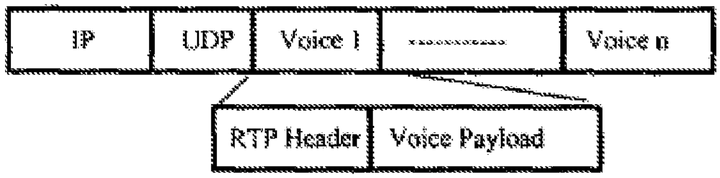


Fig. 5 Multiplexing Format

Hoshi teaches that "this method is a general RTP packet multiplexing method that is applicable not only to an IP-GW but also to other multiplexing applications." (Appx. I (Hoshi), p. 183.)

b. Overview of Rosenberg

"An RTP Payload Format for User Multiplexing" by Rosenberg, et al. ("Rosenberg"; Appx. J) is an IETF Internet Draft in the AVT Working Group dated May 6, 1998. (*See* Appx. J (Rosenberg).) Rosenberg was published prior to the filing date of the '858 patent. (*See* Appx. I (Hoshi), 188 (citing Rosenberg); Appx. BB (Internet Archive), 2 (www.ietf.org/ids.by.wg/avt.html web page listing Rosenberg and archived on February 9, 1999); Appx. CC (Internet Archive), 1 (http://www.ietf.org/internet-drafts/draft-ietf-avt-aggregation-00.txt web page for Rosenberg archived January 28, 1999).) Rosenberg was also cited in at least the following patents, having filing dates prior to the '858 patent:

- U.S. Patent No. 6,993,021 (Appx. V), filed March 8, 1999;
- U.S. Patent No. 6,704,311 (Appx. W), filed June 25, 1999;
- U.S. Patent No. 6,542,504 (Appx. X), filed May 28, 1999; and

- U.S. Patent No. 6,804,237 (Appx. Y), filed June 23, 1999.

Rosenberg is therefore prior art under at least pre-AIA 35 U.S.C. §§ 102(a) and (b).

Rosenberg describes "an RTP payload format for multiplexing data from multiple users into a single RTP packet." (Appx. J (Rosenberg), p. 1.) Rosenberg describes its technique in the context of an Internet telephony gateway (ITG) scenario, depicted in Figure 1 below. In this example, user A "wishes to speak with user C, and B wishes to speak with user D, both of which are connected to local phone network Y." (Appx. J (Rosenberg), p. 2.) To complete the call, "ITG J packetizes and transports the voice to and from A and B through the IP network, to remote gateway K" which in turn "completes the calls to C and D through PSTN Y." (Appx. J (Rosenberg), p. 2.) Rosenberg observes "that using a separate RTP session for each user connected between a pair of gateways is wasteful" because "payloads carried in each packet are generally small." (Appx. J (Rosenberg), p. 2.)

```
      ---------      -------      ----------      -------      ---------
A --|         |    |       |    |          |    |       |    |         |-- C
    |  PSTN   |----|  ITG  |----|  IP NET  |----|  ITG  |----|  PSTN   |
B --|    X    |    |   J   |    |          |    |   K   |    |    Y    |-- D
      ---------      -------      ----------      -------      ---------

Figure 1: Internet telephony gateway architecture
```

Rosenberg therefore multiplexes the payloads from both users "into the same RTP session and packet." (Appx. J (Rosenberg), p. 2.) The format of

155853739.8
155853739.12

Rosenberg's RTP packets with multiplexed user payloads is shown in Figure 2 below. In the RTP header, the "payload type field designates the RTP packet as a multiplexed payload" and the SSRC field "is used to identify groups of users (instead of a single user) whose frames are time synchronized." (Appx. J (Rosenberg), pp. 3–4.) The multiplexing header includes user headers associated with the group of users whose media data is included in the packet. (Appx. J (Rosenberg), p.4.) The RTP packet further includes media data for the users. (Appx. J (Rosenberg), p. 5.)

```
  0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                          RTP Header                           |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                          mux header                           |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       user 1 media data                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       user 2 media data                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       user 3 media data                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                                ......
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       user N media data                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

c. Motivation to combine

A POSITA would have been motivated to use RTP multiplexing to transport audio data for a conference between a conferencing bridge (e.g., Botzko's bridge

155853739.8
155853739.12

12) and a transport element in the LAN in the combination of Botzko and Kumar. Although both Hoshi and Rosenberg describe their techniques in the context of transport between Internet gateways, Hoshi suggests the use in other network elements teaching that RTP packet multiplexing "is applicable not only to an IP-GW but also to other RTP multiplexing and de-multiplexing applications." (Appx. I (Hoshi), p. 182; *see also id*., p. 183.) Rosenberg similarly "describes an RTP payload format for multiplexing data from multiple users into a single RTP packet. (Appx. J (Rosenberg), Abstract; *see also*, Conclusion ("This document has specified an RTP payload format allowing multiple user media frames to reside in an RTP packet").)

A POSITA would be further motivated to use RTP packet multiplexing in the combination of Botzko and Kumar to reduce header overhead, a benefit stressed by both Hoshi and Rosenberg. Prior to the filing date of the '858 patent, header overhead was recognized as "a key issue for voice streams in IP telephony." (Appx. I (Hoshi), p. 183.) Because of the size of the header to payload for voice packets, "only one-third of all the data is useful data (payload) and other two-thirds is overhead" making the current VoIP transfer "very inefficient." (Appx. I (Hoshi), p. 183.) Rosenberg provides an example of an ITU G.729 speech coder that "generates a rate of 8 kb/s in frames of 10 ms duration." (Appx. J (Rosenberg), p. 2) "If packed three frames per packet, the resulting RTP payloads are 30 bytes

long" and "the IP, UDP and RTP headers add up to 40 bytes, resulting in a packet efficiency of only 43%." (Appx. J (Rosenberg), p. 2) With RTP multiplexing, "the efficiency improves to 59%." (Appx. J (Rosenberg), p. 2)

A POSITA would have been further motivated to use RTP multiplexing in the combination of Botzko and Kumar to reduce packetization delays. (Appx. J (Rosenberg), p. 2) "Most Internet telephony applications use fairly large packetization delays, mainly for the purpose of raising the size of the payloads to increase efficiency." (Appx. J (Rosenberg), pp. 2–3.) With RTP multiplexing, "the packet payload increases" allowing "smaller packetization delays to be used as the number of multiplexed users increases." (Appx. J (Rosenberg), p. 2.)

A POSITA would have been further motivated to use RTP multiplexing to reduce interrupt processing at the network element. (Appx. J (Rosenberg), p. 3.) Whenever a packet arrives at the gateway, "the operating system must perform a context switch into the kernel and process the packet." (Appx. J (Rosenberg), p. 3.) The "frequency of these interrupts increases linearly with the number of users." (Appx. J (Rosenberg), p. 3.) However, with RTP multiplexing, "the packet rate does not increase as more users are added, and thus the interrupt rate stays constant … improv[ing] scalability." (Appx. J (Rosenberg), p. 3.)

2. Ground XI: The combination of Botzko, Kumar and Hoshi renders claims 1-3, 5-8, and 10 obvious.

a. The combination of Botzko, Kumar, and Hoshi renders independent claims 1 and 6 obvious.

For the reasons discussed in Ground X, the combination of Botzko and Kumar teaches limitations [1P]–[1.4] and [1.7]–[1.9] of claim 1, limitations [6P]–[6.3] and [6.5]-[6.7] of claim 6, as well dependent claims 2-3 and 7-8 (*See supra* §§V.E.2 & V.E.3.) For ease of discussion, the analyses of the limitations [1P]–[1.4], [1.6]-[1.9], [6P]–[6.3], and [6.5]-[6.7] and claims 2, 3, 7, and 8 are not repeated in this section. Hoshi discloses the "*multiplexing*"/"*multiplexor*" limitations [1.5]/[6.4] under Requestor's interpretation of the plain and ordinary meaning of "*multiplexed stream*."

As discussed in §II.D.1 (Claim Construction), the plain and ordinary meaning of "*multiplexed stream*" under Requestor's interpretation is "a data structure containing a continuous sequence of interleaved packets of audio data from each client on the active speakers list." The combination of Botzko, Kumar, and Hoshi discloses "*(5) multiplexing said packets of audio data received from each client on said active speakers list into a multiplexed stream*" [1.5] and "*a multiplexor capable of multiplexing said packets of audio data received from each client on said list of active speakers into a multiplexed stream*" [6.4] under this interpretation. (Appx. C (Bress Decl.), ¶¶254–60.)

As discussed in Ground X, the selector in the bridge of the combined system of Botzko and Kumar selects "*packets of audio data received from each client on said active speakers list*" to send to SITES with local mixing capabilities. (Appx. C (Bress Decl.), ¶255.) As discussed in Section V.E.2.f ([1.6]), each SITE "receives time compressed audio packets, here, for example, through an RTP/RTCP transport." (Appx. M (Botzko), 4:35-36.)

Hoshi discloses two techniques for voice stream multiplexing using RTP transport (like disclosed in Botzko). In the first technique, "multiplexing is done in the RTP layer so that voice frames from each voice stream are multiplexed and encapsulated into an RTP packet." (Appx. I (Hoshi), p. 184.) In the second technique, "multiplexing RTP voice packets into a UDP frame, the multiplexing layer is between the RTP and UDP layers." (Appx. I (Hoshi), p. 184.) Specifically, each input stream "is sampled at a multiplexing interval by RTP packet basis, which is independent from packetizing interval timing, and sampled RTP packets are concatenated into a multiplexed packet, which is sent to the destination IP-GW by adding UDP and IP headers." (Appx. I (Hoshi), p. 185.)

In the combined system of Botzko and Kumar, utilizing Hoshi's voice stream multiplexing, the RTP/RTCP transport circuit multiplexes packet streams from each of the active speaker clients at a multiplexing interval. (Appx. C (Bress Decl.), ¶259.) The length of the multiplexed packet "is variable, depending on the

155853739.8
155853739.12

types of codecs and the number of RTP packets concatenated." (Appx. I (Hoshi), p. 185.) Thus, the resulting multiplexed packet (illustrated in Hoshi Figure 5) is a data structure that contains a continuous sequence of interleaved packets of audio data from each client on the active speakers list. (Appx. C (Bress Decl.), ¶259.)

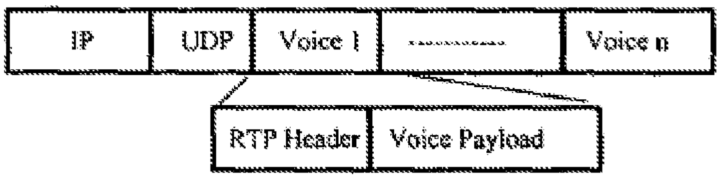


Fig. 5 Multiplexing Format

The combination of Botzko, Kumar, and Hoshi therefore discloses or at least suggests "*(5) multiplexing said packets of audio data received from each client on said active speakers list into a multiplexed stream*" [1.5] under Requestor's interpretation. (Appx. C (Bress Decl.), ¶¶254–59.)

As explained in Section V.E.3.e, a POSITA would understand that, in the combination of Botzko and Kumar, the RTP/RTCP transport circuit in Botzko's bridge includes a "*multiplexor capable of*" the disclosed "*multiplexing*." A POSITA would understand that in the combination of Botzko, Kumar, and Hoshi, the RTP/RTCP transport circuit would include a component for performing Hoshi's voice stream multiplexing. (Appx. C (Bress Decl.), ¶260.) The combination of Botzko, Kumar, and Hoshi therefore also discloses or at least suggests "*a multiplexor capable of multiplexing said packets of audio data*

received from each client on said list of active speakers into a multiplexed stream" [6.4] under Requestor's interpretation of "*multiplexed stream*." (Appx. C (Bress Decl.), ¶¶254–60.)

b. The combination of Botzko, Kumar, and Hoshi renders dependent claims 5 and 10 obvious.

The combination of Botzko, Kumar, and Hoshi discloses that "*at least one of said second subset of the plurality of clients is using a telephone and the H.323 protocol*" as recited in dependent claim 5 and that "*at least one of the plurality of clients, which does not have the capability to mix multiple audio streams, is using a telephone and the H.323 protocol*" as recited in dependent claim 10. (Appx. C (Bress Decl.), ¶¶261–65.)

Botzko discloses that its SITES "can connect to the bridge in many different ways, for example, through the **public switched telephone network**, by wireless, by direct connection, or in any other desired combination of the various communication paths including, for example, a local area network." (Appx. M (Botzko), 4:18-24.) A POSITA would understand that a PSTN SITE would connect to a packet-based network via a gateway. (Appx. C (Bress Decl.), ¶263 *citing* Appx. G (Pearce), 5:9-24; Figure 1; Appx. H (Oran), 2:54-56, Figure 1; Appx. K (Robert), Figure 3.) The SITE coupled to the PSTN is a "*telephone*." (Appx. A ('858 patent), 3:66-4:1 (equating telephone-based clients to clients that connect to the PSTN).)

Botzko's conferencing system supports normal end-points "which can receive one audio stream to play out of their loudspeakers." (Appx. M (Botzko), 5:23-25.) A POSITA would have understood that one or more terminals coupled to the PSTN via a gateway in the combination of Botzko and Kumar are normal end-points (i.e., non-mixing end-points.)

Hoshi explains that its IP telephony system "is based on many standards such as ITU-T H.323 packet-based multimedia communication systems." (Appx. I (Hoshi), p. 182.) Hoshi notes that an "advantage of this method is that no new additional headers are required and current well-defined H.323 and RTP standards can be applied." (Appx. I (Hoshi), p. 183.) A POSITA would therefore understand that telephones (Botzko's non-mixing SITES) coupled to a PSTN gateway communicate with Botzko's bridge communicate using H.323 in the same manner described in the '858 patent. (Appx. C (Bress Decl.), ¶265.) Indeed, the '858 patent acknowledges that H.323 was adopted more than 4 years before its filing date. (*See* Appx. A ('858 patent), 1:47-59.)

Accordingly, the combination of Botzko, Kumar, and Hoshi discloses that "*at least one of said second subset of the plurality of clients is using a telephone and the H.323 protocol* [via a gateway]" [5] and that "*at least one of the plurality of clients, which does not have the capability to mix multiple audio streams, is*

155853739.8
155853739.12

using a telephone and the H.323 protocol [via a gateway]" [10] in the same manner as the '858 patent. (Appx. C (Bress Decl.), ¶¶261–65.)

3. GROUND XII: The combination of Botzko, Kumar, and Rosenberg renders claims 1–10 obvious.

a. The combination of Botzko, Kumar, and Rosenberg renders independent claims 1 and 6 obvious.

For the reasons discussed in Ground X, the combination of Botzko and Kumar discloses limitations [1P]–[1.4] and [1.7]–[1.9] of claim 1 and limitations [6P]–[6.3] and [6.5]-[6.7] of claim 6 and claims 2, 3, 7, and 8. (*See supra* §§V.E.2 & V.E.3.) For ease of discussion, the analyses of the limitations [1P]–[1.4], [1.6]-[1.9], [6P]–[6.3], and [6.5]-[6.7] and claims 2, 3, 7, and 8 are not repeated in this section.

The combination of Botzko, Kumar, and Rosenberg discloses "*(5) multiplexing said packets of audio data received from each client on said active speakers list into a multiplexed stream*" [1.5] and "*a multiplexor capable of multiplexing said packets of audio data received from each client on said list of active speakers into a multiplexed stream*" [6.4] under Requestor's interpretation of "*multiplexed stream*." (Appx. C (Bress Decl.), ¶¶266–71.)

As discussed in Ground X, the selector in the bridge of the combined system of Botzko and Kumar selects "*packets of audio data received from each client on said active speakers list*" to send to SITES with local mixing capabilities. (Appx. C

(Bress Decl.), ¶267.) As discussed in Section V.E.2.f ([1.6]), each SITE "receives time compressed audio packets, here, for example, through an RTP/RTCP transport." (Appx. M (Botzko), 4:35-36.)

Rosenberg teaches that the transmission is via a "data structure." (Appx. C (Bress Decl.), ¶268.) Rosenberg describes "an RTP payload format for multiplexing data from multiple users into a single RTP packet." (Appx. J (Rosenberg), p. 1.) The format of Rosenberg's RTP packets with multiplexed users is shown in Figure 2 below. The RTP packet includes an RTP header with a payload type field designating "the RTP packet as a multiplexed payload" and an SSRC field identifying groups of users "whose frames are time synchronized", a multiplexing header, and media data for the users in the group of users. (Appx. J (Rosenberg), pp. 3–5.)

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                        RTP Header                             |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                        mux header                             |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     user 1 media data                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     user 2 media data                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     user 3 media data                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                                ......
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     user N media data                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

In the combined system of Botzko and Kumar, utilizing Rosenberg's RTP multiplexing, the RTP/RTCP transport circuit multiplexes packet streams from each of the active speaker clients at an audio sampling rate (as discussed above). (Appx. C (Bress Decl.), ¶270.) Thus, the resulting multiplexed packet (illustrated in Rosenberg's Figure 2) is a data structure that contains a continuous sequence of interleaved packets of audio data from each client on the active speakers list. (Appx. C (Bress Decl.), ¶270.)

The combination of Botzko, Kumar, and Rosenberg therefore discloses or at least suggests "*(5) multiplexing said packets of audio data received from each client on said active speakers list into a multiplexed stream*" [1.5] under Requestor's interpretation. (Appx. C (Bress Decl.), ¶¶266–70.)

As explained in Section V.E.3.e, a POSITA would understand that, in the combination of Botzko and Kumar, the RTP/RTCP transport circuit in Botzko's bridge includes a "*multiplexor capable of*" the disclosed "*multiplexing.*" A POSITA would understand that in the combination of Botzko, Kumar, and Rosenberg, the RTP/RTCP transport circuit would include a component for performing Rosenberg's RTP multiplexing. (Appx. C (Bress Decl.), ¶271.) The combination of Botzko, Kumar, and Rosenberg therefore also discloses or at least suggests "*a multiplexor capable of multiplexing said packets of audio data received from each client on said list of active speakers into a multiplexed stream*" [6.4] under Requestor's interpretation. (Appx. C (Bress Decl.), ¶¶266–71.)

b. The combination of Botzko, Kumar, and Rosenberg renders dependent claims 4 and 9 obvious.

The combination of Botzko, Kumar, and Rosenberg discloses that "*at least one of said first subset of the plurality of clients is using PC-based equipment and the Session Initiation Protocol (SIP)*" as recited in dependent claim 4 and that "*at least one of the plurality of clients, which has the capability to mix multiple audio streams, is using PC-based equipment and the Session Initiation Protocol (SIP)*" as recited in dependent claim 9. (Appx. C (Bress Decl.), ¶¶272–74.)

Botzko teaches that its SITES "can connect to the bridge in many different ways, for example, through the public switched telephone network, by wireless, by direct connection, or in any other desired combination of the various

communication paths including, **for example, a local area network**." (Appx. M (Botzko), 4:18-23.) Botzko further teaches that "[s]ome end-point SITES are able to receive more than one audio stream, and perform their own local mixing." (Appx. M (Botzko), 5:39-40.) These local mixing end-point SITES are "*PC-based equipment*." (*See* Appx. A ('858 patent), 3:58-63 (equating PC-based clients to clients which connect to a packet-switched network such as a WAN or the Internet), 2:18-22 (equating PC-based clients to clients that have the capability to mix).) Thus, the combination of Botzko and Kumar discloses "*at least one of said first subset of the plurality of clients is using PC-based equipment*" [4] and "*at least one of the plurality of clients, which has the capability to mix multiple audio streams, is using PC-based equipment*" [9]. (Appx. C (Bress Decl.), ¶273.)

Rosenberg notes that its gateway devices "can signal calls using SIP [1], H.323 or proprietary signalling [sic] protocols." (Appx. J (Rosenberg), p. 2.) Rosenberg also teaches that "signaling protocols such as SIP [6] together with a session description protocol such as SDP or H.323 could be used" to bind payload types to particular codec types and length values. (Appx. J (Rosenberg), p. 5.) Based on Rosenberg's disclosure, a POSITA would understand that SIP would also be used by IP telephony SITES of Botzko for signaling. (Appx. C (Bress Decl.), ¶274; *see also* Appx J (Rosenberg), p. 8 ("The multiplexing protocol can make use of whatever encryption and authentication schemes are present in RTP, SIP, H.323

or other relevant protocols").) Indeed, the '858 patent acknowledges that SIP was a well-known "signaling protocol for Internet conferencing and telephony" prior to its filing date. (*See* Appx. A ('858 patent), 1:61-66.)

Accordingly, the combination of Botzko, Kumar, and Rosenberg discloses that "*at least one of said first subset of the plurality of clients is using PC-based equipment and the Session Initiation Protocol (SIP)*" [4] and that "*at least one of the plurality of clients, which has the capability to mix multiple audio streams, is using PC-based equipment and the Session Initiation Protocol (SIP)*" [9]. (Appx. C (Bress Decl.), ¶¶272–74.)

c. The combination of Botzko, Kumar, and Rosenberg renders dependent claims 5 and 10 obvious.

The combination of Botzko, Kumar, and Rosenberg discloses that "*at least one of said second subset of the plurality of clients is using a telephone and the H.323 protocol*" as recited in dependent claim 5 and that "*at least one of the plurality of clients, which does not have the capability to mix multiple audio streams, is using a telephone and the H.323 protocol*" as recited in dependent claim 10. (Appx. C (Bress Decl.), ¶¶275–78.)

Botzko discloses that its SITES "can connect to the bridge in many different ways, for example, through the **public switched telephone network**, by wireless, by direct connection, or in any other desired combination of the various communication paths including, for example, a local area network." (Appx. M

(Botzko), 4:18-24.) A POSITA would understand that a PSTN SITE would connect to a packet-based network via a gateway. (Appx. C (Bress Decl.), ¶276 *citing* Appx. G (Pearce), 5:9-24; Figure 1; Appx. H (Oran), 2:54-56, Figure 1; Appx. K (Robert), Figure 3.) A SITE coupled to the PSTN is a "*telephone*." (Appx. A ('858 patent), 3:66-4:1 (equating telephone-based clients to clients that connect to the PSTN).)

Botzko's conferencing system supports normal end-points "which can receive one audio stream to play out of their loudspeakers." (Appx. M (Botzko), 5:23-25.) A POSITA would have understood that one or more telephones coupled to the PSTN via a gateway in the combination of Botzko and Kumar are normal end-points (i.e., non-mixing endpoints.) The combination of Botzko and Kumar therefore discloses that "*at least one of said second subset of the plurality of clients is using a telephone*" [5] and "*at least one of the plurality of clients, which does not have the capability to mix multiple audio streams, is using a telephone*" [10]. (Appx. C (Bress Decl.), ¶277.)

Rosenberg notes that its gateway devices "can signal calls using SIP [1], H.323 or proprietary signalling [sic] protocols." (Appx. J (Rosenberg), p. 2.) Rosenberg also teaches that "signaling protocols such as SIP [6] together with a session description protocol such as SDP or H.323 could be used" to bind payload types to particular codec types and length values. (Appx. J (Rosenberg), p. 5.) A

POSITA would therefore understand that the gateway, IP telephones, multicast gateway, and MCU communicate using H.323 in the combination of Botzko, Kumar, and Rosenberg. (Appx. C (Bress Decl.), ¶278.) Indeed, the ’858 patent acknowledges that H.323 was adopted more than 4 years before its filing date. (*See* Appx. A (’858 patent), 1:47-59.) Rosenberg therefore teaches that “*at least one of said second subset of the plurality of clients is using … the H.323 protocol* [via a gateway]” [5] and “*at least one of the plurality of clients, which does not have the capability to mix multiple audio streams, is using … the H.323 protocol* [via a gateway]” [10] in the same manner as discussed in the ’858 patent. (Appx. C (Bress Decl.), ¶¶275–78.)

VI. CONCLUSION

For the reasons above, the references discussed herein raise substantial and new questions of patentability with respect to claims 1–10 of the ’858 patent. These references were filed or published prior to the filing date of the ’858 patent. The references and set forth in Grounds I–XII render claims 1–10 unpatentable for the reasons discussed above.

Respectfully submitted,

/Lori A. Gordon/

Lori A. Gordon, Reg. No 50,633
Attorney for Third Party Requester,
Cisco Systems, Inc.

Date: February 22, 2002

Perkins Coie LLP
P.O. Box 1247
Seattle, Washington 98111-1247
Telephone: (858) 720-5700
Facsimile: (206) 359-7198